# NAVAL POSTGRADUATE SCHOOL
## Monterey, California

# THESIS

# NAVAL POSTGRADUATE SCHOOL
## Monterey, California

# THESIS

COMPUTER SIMULATION OF
DIGITAL SIGNAL MODULATION TECHNIQUES IN
SATELLITE COMMUNICATIONS

by

Craig Dean Carlson

September 1985

Thesis Advisor: James L. Wayman

Approved for public release; distribution is unlimited

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

## REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | | |
|---|---|---|
| 1. REPORT NUMBER | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE *(and Subtitle)* Computer Simulation of Digital Signal Modulation Techniques in Satellite Communications | | 5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1985 |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s) Craig D. Carlson | | 8. CONTRACT OR GRANT NUMBER(s) |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100 | | 12. REPORT DATE September 1985 |
| | | 13. NUMBER OF PAGES 291 |
| 14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* | | 15. SECURITY CLASS. *(of this report)* |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution is unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Computer Simulation, Digital Signal, Modulation Techniques, Satellite Communications, Statistical Analysis

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This thesis is a tutorial on digital signal modulation techniques used in satellite communications and includes computer simulation of those digital signal modulation techniques introduced. The purpose of the thesis is to introduce digital signal modulation techniques and through the use of computer simulation, generate statistics which represent the characteristics of the FFT for the respective signal type. Further, an analysis of the statistics

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601



SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)* of the FFT's was conducted to determine if there is any relationship between the components of the FFT of the different signals. The statistic used to investigate this possible relationship was the F-distribution. The computer simulation was written and conducted in the FORTRAN programming language. A copy of the program, results of the simulation and the statistical analysis conducted are included in the appendices.

# Computer Simulation of Digital Signal Modulation Techniques in Satellite Communications

by

Craig D. Carlson
Lieutenant Commander, United States Navy
B.A., Concordia College, 1973

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(SPACE SYSTEMS OPERATIONS)

from the

NAVAL POSTGRADUATE SCHOOL
September 1985

## ABSTRACT

This thesis is a tutorial on digital signal modulation techniques used in satellite communications and includes computer simulations of those digital signal modulation techniques introduced. The purpose of the thesis is to introduce digital signal modulation techniques and through the use of computer simulation, generate statistics which represent the characteristics of the FFT for the respective signal type. Further, an analysis of the statistics of the FFT's was conducted to determine if there is any relationship between the components of the FFT of the different signals. The statistic used to investigate this possible relationship was the F-distribution. The computer simulation was written and conducted in the FORTRAN programming language. A copy of the program, results of the simulations and the statistical analysis conducted are included in the appendices.

TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# I. INTRODUCTION

## A. BACKGROUND

Since the introduction of the sampling theorem and the matched filter, digital communications techniques have developed into a highly proficient discipline. The marriage of this discipline with the rapidly expanding space program has resulted in communication satellites employing a multitude of digital signal modulation techniques. A modulation technique is a method of transmitting the information contained in a message by varying or modulating the characteristics of a carrier waveform. These methods offer a range of advantages and disadvantages depending on the specific characteristics of the modulation technique employed. The applications of the various signal modulation techniques likewise vary. In order to understand this exciting new field it is necessary to look at some of the aspects of satellite communications systems in general. Then an investigation will be made into the general attributes of digital communications and their relationship to the satellite system.

The ability to understand these digital signal modulation techniques is the first step in being able to intercept, identify and demodulate unknown digital signals. These digital signals, transmitted from unknown sources, are believed to display frequency characteristics peculiar to the modulation technique employed in the encoding process. The digital computer offers unique opportunities in simulating these modulation techniques, in signal processing and in decoding of an intercepted signal.

## B. SPECIFIC GOALS

It is the specific goal of this thesis to investigate the open literature on the topic of digital signal modulation techniques in satellite communications. This includes a basic understanding of the communications satellite system as well as the specific techniques employed in signal modulation. Additionally, once a basic understanding of these digital signal modulation techniques is achieved, computer code in the FORTRAN programming language will be developed which simulates these modulated signals. The statistics of the time-varying Fast Fourier Transforms (FFT) of these simulated signals will be investigated. The purpose of this analysis will be to lead to follow-on research in the area of signal analysis of intercepted digital signals whereby they can be classified by their FFT as employing a specific digital signal modulation technique.

## C. SCOPE OF THE PROJECT

Although an indepth analysis of all the digital signal modulation techniques which will be introduced involves considerable higher level mathematics, it is not within the scope of this project to delve heavily into the mathematics. It would be advantageous, however, for the reader to have had integral and differential calculus and an introduction to statistics. Also a course in electrical engineering may prove helpful but is not essential since in actuality it is the electrical engineering aspects of satellite communications that this tutorial is attempting to present. The computer programming will be accomplished utilizing the techniques of software engineering and top down modular design. Existing blocks of code will be used as modules whenever appropriate and available. Again it is the ultimate purpose of this project to examine a variety of digital signal modulation techniques and develop computer code that simulates common signals used in satellite communications that have been produced by one of the many digital signal modulation techniques to be investigated.

## II. UNDERSTANDING SATELLITE COMMUNICATIONS

### A. HISTORY AND APPLICATIONS

Forty years ago, in 1945, Arthur Clarke first envisioned the use of space stations placed in geosynchronous orbit for communicating to different points on the earth [Ref. 1]. Nine years later in 1954, J.R. Pierce of Bell Laboratories performed a system analysis on such a communications system [Ref. 2]. In 1957, the launch of Sputnik demonstrated the feasibility of using a satellite for just such an application. However, by 1961 the only satellite communications technologies which had been demonstrated were the Courier 1B satellite, a short-life active retransmission teletype communications satellite in a 1000 km orbit, and the Echo I balloon in a 1600 km orbit which demonstrated passive reflection of powerful microwave signals from one earth station to another. Active microwave communications demonstrated in Projects Telstar and Relay were years away [Ref. 3]. The first geostationary orbit was achieved by Project Syncom in 1963 [Ref. 4].

In 1964, communication organizations from several countries joined together to form the international organization of INTELSAT (International Telecommunications Satellite Organization). INTELSAT's purpose was to develop a satellite network which would provide truly global communications capabilities. This resulted in the launch in 1965 of the world's first commercial communications satellite, INTELSAT I, also known as "Early Bird". With "Early Bird", telecommunications utilizing satellite relay were established between the United States and Europe. [Ref. 3]

The reliability of these satellite systems has improved dramatically since INTELSAT I in 1965. Reliability of individual links in the system approach 99.99 percent or higher. Total system reliability exceeding 99.9 percent is common [Ref. 5], making satellite communications more reliable than many other modes of communications. In this sense, reliability is a measure of the probability that no failure will occur in a respective channel or in the system during the design life of the satellite.

In order for the operability, capability and reliability of these systems to have developed at this pace, it was necessary for the technologies associated with them to develop as rapidly or even more rapidly than the systems themselves. Engineering sciences and specifically the fields of aerospace and electrical engineering historically have required between 7 and 10 years to take a concept from operational requirement to full scale operation. This was not the case with the concept of communications satellites which has seen four generations of INTELSAT satellites within a decade's time. [Ref. 3]

The "Early Bird" satellite weighed approximately 38 kilograms and possessed limited power and bandwidth capacity enabling it to carry only 240 two-way telephone conversations. Today's communications satellites represent order-of-magnitude improvements in many important operating parameters such as power and bandwidth. Additionally, increased effective radiated power from the techniques of stabilized earth pointing antennas have greatly increased the capacity of later generation communications satellites. INTELSAT V, the current generation of communications satellites, is a three-axis stabilized platform using not only the 6/4 GHz frequency band (6 GHz receive/4 GHz transmit) as in earlier generation satellites but also a 500 MHz bandwidth available in the 14/11 GHz band. The separation in the receive and transmit frequencies is necessary to prevent interference during simultaneous operation of the receiver and transmitter. Another factor increasing the capacity of present generation satellites is the increase in primary power available in the satellite. [Ref. 3]

The technologies which have contributed to the evolution of communications satellites come from two primary sources, namely technology from the space program of the 1960's supported by NASA and DoD and communications technology due largely to commercial and private sector contributions. [Ref. 3]

Electronic devices and components have contributed significantly to the rapid development of satellite systems. These electronic devices range from something which is now considered basic, i.e., the transistor, to devices such as Traveling Wave Tube amplifiers, lightweight antennas and antenna feed systems. There have been major improvements in satellite power sources including more efficient solar cells and high storage capacity/lightweight batteries. Additionally, two significant developments in electrical engineering have made modern day digital signal processing a reality. They are the sampling theorem and the matched filter. [Ref. 3]

Communications satellites of the future are likely to utilize onboard signal processing. Signal processing functions such as signal reshaping, switching and/or multiplexing and compression could soon take place on the satellite due to the reduction in size and weight of the necessary hardware. Operation at over 100 megabits per second are envisioned. Onboard signal processing will greatly reduce the expense and complexity of earth stations thereby making services of a satellite available to a wider range of small and geographically disperse users. [Ref. 3]

Satellite communications hold great promise to provide service to a multitude of users over a wide area. Applications lie not solely in retransmission but also in data collection from that same large geographic area. Military users have definite applications in these areas when warning, intelligence and surveillance systems provide digital inputs to a central source. Although most of the present applications for satellite communications are still provided by the government, commercial applications have seen tremendous increases in the last several years. [Ref. 3]

## B. ORBITS AND LIMITATIONS

Satellite orbits are generally categorized as either equatorial (0 degrees inclination), polar (90 degrees inclination) or inclined at some angle other than 0 or 90 degrees relative to the spin axis of the earth as illustrated in Figure 2.1 [Ref. 6]. Each satellite orbit has a characteristic velocity which is dependent on the height of the orbit and the orbit's eccentricity. Eccentricity is a measure of the degree to which the orbit approximates a circle. A circle has eccentricity equal to zero since the focus is located at the center. See equation 2.1 and Figure 2.2.

A communications satellite in elliptical orbit about the earth obeys Kepler's second law of planetary motion. That is, a satellite's constant angular momentum about the earth means that its areal velocity also remains constant. See Figure 2.3. Of particular interest is the satellite velocity at apogee (Va) and perigee (Vp) given by equations 2.2 and 2.3.

Note that for a circle, $e = 0$ and $Ra = Rp$. Therefore a satellite in circular orbit about the earth has an orbital velocity as given in equation 2.4.

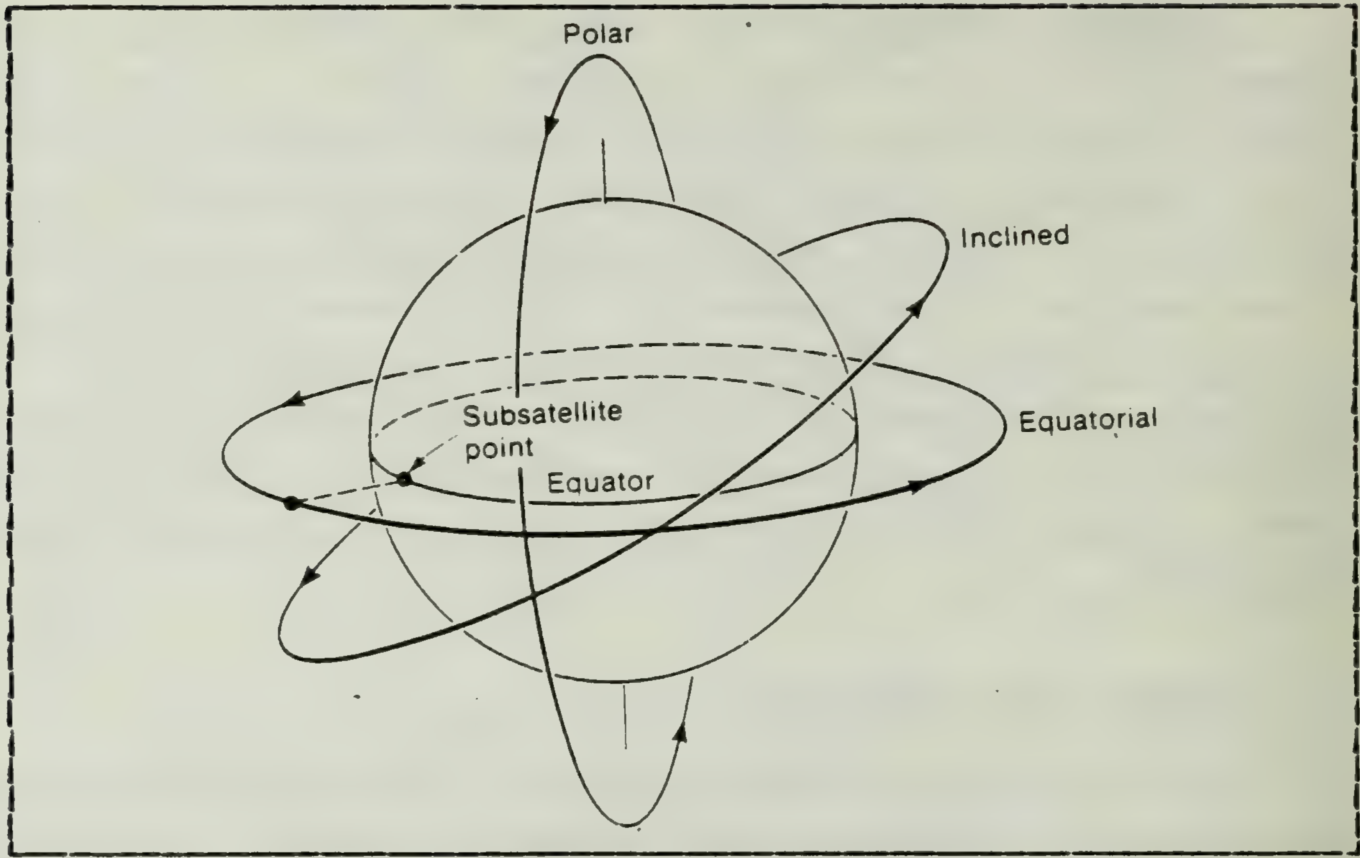



**Figure 2.1 Satellite Orbits**

Most communications satellites are placed in a circular equatorial orbit where the desire is to stabilize them over a fixed point on the surface of the earth called the subsatellite point. This type of orbit is called geostationary or stationary. Any satellite which has an orbital period equal to the period of rotation of the earth is called synchronous or geosynchronous. These terms are used almost interchangeably in most literature. As mentioned for a geosynchronous communications satellite, the orbital period of the satellite, T, must be equal to the period of rotation of the earth. Kepler's third law of planetary motion can be rewritten to yield equation 2.5.

The period of revolution of the earth for a sidereal day is 23 hr 56 min 4 sec. For that given period there is only one satellite altitude as expressed by Kepler's third law.

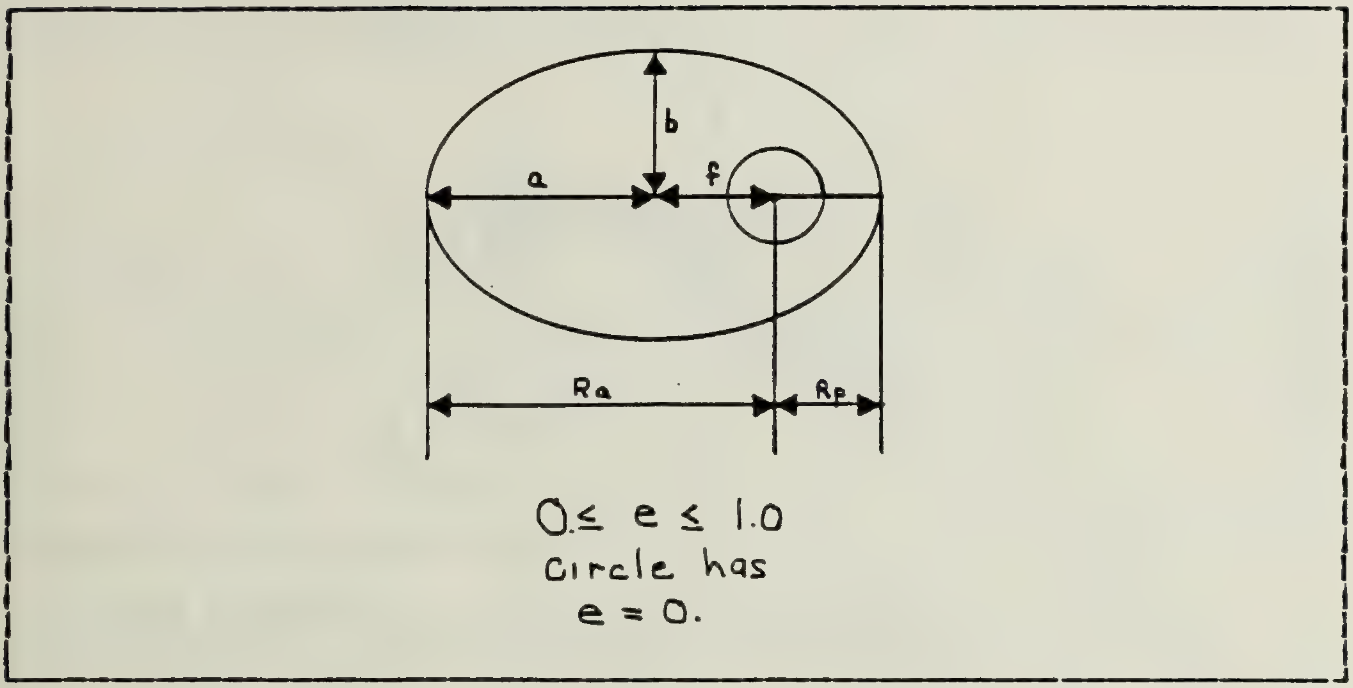



**Figure 2.2 Eccentricity**

eccentricity (e) = c/a (eqn 2.1)

c = distance from focus
a = semi-major axis

b = semi-minor axis
Ra = radius of apogee
Rp = radius of perigee

By rearranging terms that altitude is given in equation 2.6. Since the orbit is circular, a = Ra = Rp and the height of the orbit above the surface of the earth is h = a - Re; or 35,804 km.

One disadvantage of a geosynchronous communications satellite is the lack of global coverage. These satellites provide excellent coverage of the subsatellite point and

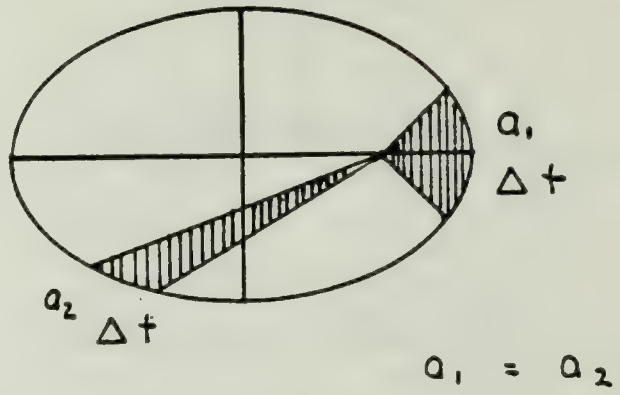



**Figure 2.3 Kepler's Second Law of Planetary Motion**

$$Va = (k/Ra(1-e))^{.5} \qquad \text{(eqn 2.2)}$$

$$Vp = (k/Rp(1+e))^{.5} \qquad \text{(eqn 2.3)}$$

$k = GMe = 3.98866 \times 10^{-11}\ m^3/s^2$

$G$ = universal gravitational constant

$G = 6.67 \times 10^{-11}\ N\ m^2/kg^2$

$N$ = newtons = $m\ kg/s^2$

$Me$ = mass of the earth = $5.98 \times 10^{24}$ kg

$$Vc = (k/Rc)^{.5} \qquad \text{(eqn 2.4)}$$

$Rc = Re + h$

$Re$ = radius of the earth = $6.37 \times 10^{6}$ m

$h$ = satellite orbital altitude

$$T = 2\pi a^{1.5} / k^{.5} \qquad \text{(eqn 2.5)}$$

$T$ = period of rotation

$a$ = semi-major axis

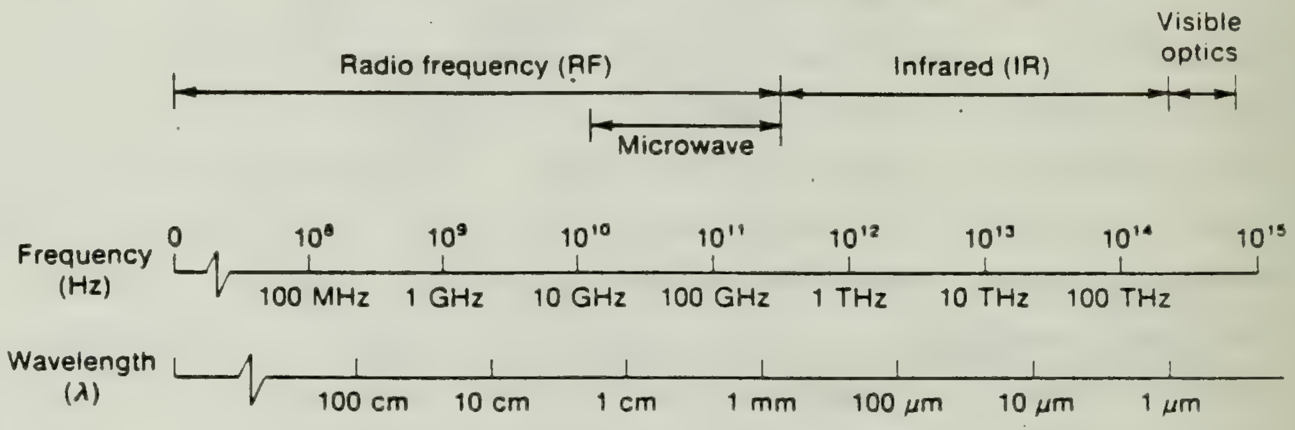
$$a = (T / 2\pi)^{2/3} (k)^{1/3} \quad \text{(eqn 2.6)}$$

$$a = 42{,}173 \text{ km}$$

laterally to a latitude of about ±80° [Ref. 6]. This is generally no problem for commercial applications since the polar regions do not require a significant degree of access. The military, on the other hand, does have an interest in communications in the polar region and therefore has several communications satellites that have orbits inclined at various angles. This type of orbit generally has disadvantages of lack of continuous coverage and a much more complicated system of ground tracking and receiving stations.

## C. FREQUENCY BAND CONSIDERATIONS

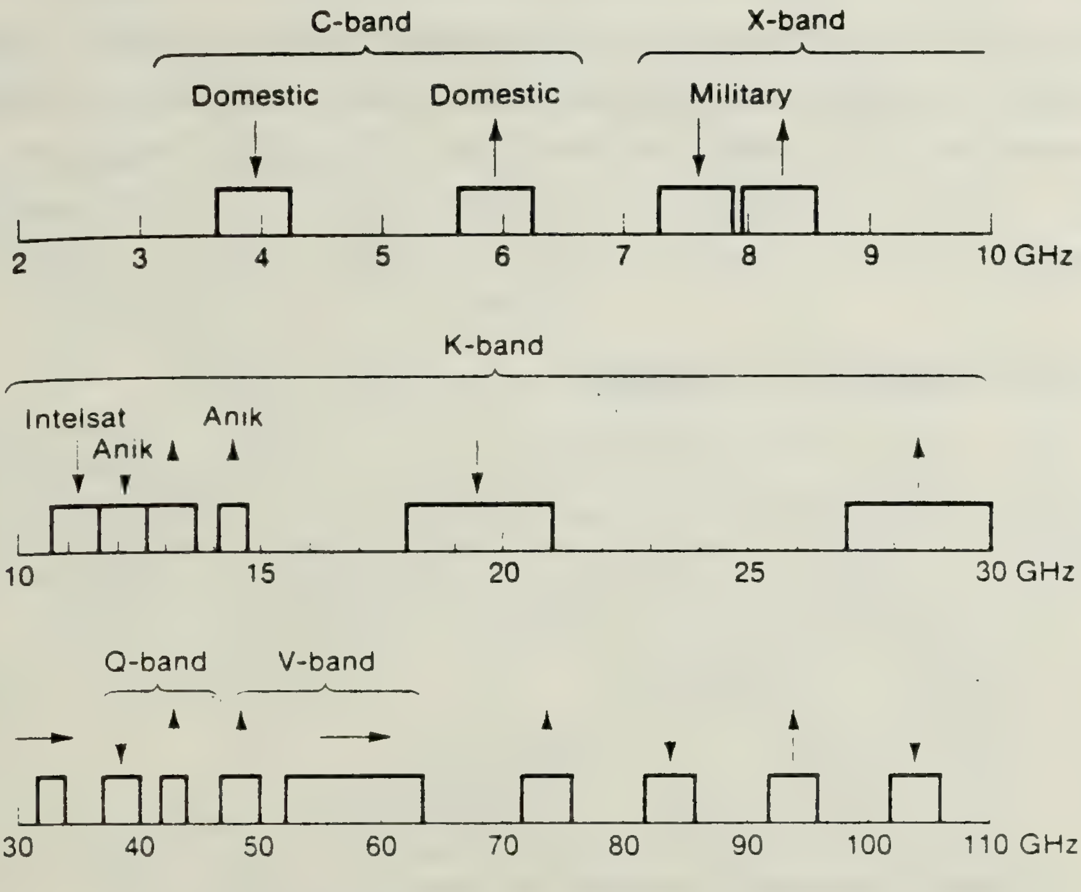
Although there appears to be an infinite number of frequencies available for communications, restrictions do exist as to those which are practicable. Limitations on available frequency bands for satellite communications are due to the need to select segments of the electromagnetic spectrum which reduce noise and interference and are most favorable in terms of power efficiency and propagation distortions. Trade offs must be made to arrive at the optimum frequency for a particular application since single frequencies seldom offer the best performance for all variables. The problems arise when consideration is given to the number of users requiring the same frequency bands including terrestrial communications networks. Since the problem of interference is global, a worldwide organization has been established to assign frequency bands for various applications. This organization is called WARC, World Administrative Radio Conference [Ref. 6]. Illustrations of the portions of the electromagnetic spectrum in question and current allocations of satellite frequency bands are shown in Figures 2.4 and 2.5 [Ref. 6].




| Bands | Frequency |
|---|---|
| VHF | 54 – 216 MHz |
| UHF | 470 – 890 |
| L | .39 – 1.55 GHz |
| S | 1.55 – 5.2 |
| C | 3.9 – 6.2 |
| X | 5.2 – 10.9 |
| K | 10.9 – 36 |
| Ku | 11.7 – 14.5 |
| Ka | 17 – 31 |
| Q | 36 – 46 |
| V | 46 – 56 |

**Figure 2.4 Electromagnetic Spectrum**

In general, as long as the hardware and technology will support it, higher frequencies are more desirable and more in demand because they offer higher theoretical capacity. This is due to the fact that only a percentage of the carrier frequency is capable of actually transmitting a signal of a given bandwidth. Higher frequencies would also experience less interference with existing land and satellite systems. A further discussion of the bandwidths available as a function of frequency will be included at a




Allocated satellite frequency bands, WARC 1979 (↑ = uplink, ↓ = downlink, → = crosslink).

Figure 2.5 Satellite Frequencies

later point under the topic of actual digital signal techniques.

As an additional note, in order to assist in the ability to handle more signals in an obviously limited electromagnetic spectrum, several multiple access schemes have been developed. The most common techniques involve frequency-division multiple access (FDMA) where the allocated satellite frequency band is divided among the users into specific uplink and downlink frequencies. Next is time-division multiple access (TDMA) where the satellite frequency band is shared by all users by carefully dividing the time any one user has access. Finally there is code-division multiple access (CDMA) which involves modulating a specific address code which has been superimposed on the signal directly onto the carrier. In this manner all users share the satellite frequency band and only those receiving stations that can demodulate the address code can receive the specific signal. [Ref. 6]

## D. WHY DIGITAL MODULATION

In general, digital signal modulation techniques are superior to analog signal modulation techniques used in satellite communications for the following reasons:

1. Compatibility with digital computers
2. Economic advantages
3. High degree of flexibility
4. Less susceptible to interference
5. Quality of signal independent of transmission distance and network makeup

### 1. Compatibility with Digital Computers

Clearly one of the most distinctive advantages of digital techniques over analog techniques involves computer applications. Properly formatted digital signals can be used to represent any analog signal (more on this under the discussion of the sampling theorem). Once in the digital format these signals can be easily manipulated within the digital computer. Arithmetic operations can be applied as well as logical operations. The signal can be stored without alteration or delayed and therefore can be used to "simulate" real physical situations.

The hardware associated with digital circuits is free from drift or aging and does not require calibration. Additionally digital circuitry is compatible with present day integrated circuit technology allowing a standardized building block construction approach. The operating characteristics of these systems employing a digital computer can be changed by altering the software rather than the hardware as is the case for analog systems. Finally, the use of digital computer technology allows time multiplexing.

### 2. Flexibility and Economy

The flexibility and economy of digital satellite communications comes from the fact that more and more processing can be done onboard. This allows the uplink and downlink to be completely separated. This regenerative nature makes it possible for low error rates and high reliability through the use of digital techniques not available to analog systems. Because digital signal processing or multiplexing is less costly than for analog signals, simpler and cheaper interfaces between earth stations and terrestrial communications networks are possible. Additionally, there are reduced production costs and increased capacity associated with digital circuits. [Ref. 7]

### 3. Quality and Interference

The capability of digital systems to regenerate the signal and allow for multiple switching and signal processing without degradation in signal quality makes the digital signal basically independent of transmission distance. Multiple hops from satellite to earth station or satellite to satellite are possible without accumulation of the noise characteristic of analog systems. Additionally, digital systems are capable of operating at a signal to noise ratio of 20 dB to 30 dB as compared to analog systems requiring a much more powerful signal. [Ref. 8]

# III. INTRODUCTION TO THE FOURIER TRANSFORM, SAMPLING THEOREM, AUTOCORRELATION FUNCTION AND THE MATCHED FILTER

Essential to the understanding of digital signal modulation techniques are a few basic tools of the electrical engineer and the mathematician. These tools will be developed and elaborated on to the extent necessary to understand their applicability to the subject of digital communications. The description is not meant to be a detailed investigation of the respective topics. Where relevant, the application of the concept being described will be mentioned.

## A. FOURIER TRANSFORM

In mathematical terms, voltages can be expressed as functions of time or of frequency. It is more common to see voltages represented as a function of time as in equation 3.1.

$$v(t) = A \cos(wt) \qquad \text{(eqn 3.1)}$$

A = amplitude
w = angular frequency = 2 pi f
f = frequency = 1/T
T = period

It is important to note that both the time and frequency functions are representations of voltage and as such may be used interchangeably. The Fourier representation [v(t)] = V(f) is a voltage descriptor in the frequency domain (a function of frequency) while v(t) is a voltage descriptor in the time domain (a function of time). They are different but equivalent and either voltage descriptor may be used, depending on the concept being explored, to best represent the voltage within context.

The Fourier transform is of principle interest rather than the Fourier series since the latter is applicable only to periodic voltages. The Fourier transform is applicable to voltage pulses, random voltages and other non-periodic voltages.

DEFINITION:

$$\mathcal{F}[v(t)] = V(w) = \int_{-\infty}^{\infty} v(t)\, e^{**}{-jwt}\, dt \qquad \text{(eqn 3.2)}$$

v(t) <-------------> V(w)

time domain frequency domain

Remembering Euler's formula

$$e^{**}{-jwt} = \cos(wt) - j \sin(wt) \qquad \text{(eqn 3.3)}$$

the Fourier transform becomes

$$\mathcal{F}[v(t)] = V(w) = \int_{-\infty}^{\infty} v(t)[\cos(wt) - j \sin(wt)] \qquad \text{(eqn 3.4)}$$

$$V(w) = \int_{-\infty}^{\infty} v(t)\cos(wt) - j\int_{-\infty}^{\infty} v(t)\sin(wt) \qquad \text{(eqn 3.5)}$$

Since it will be seen that digital communications deals primarily with pulses of finite duration (expressed as period, T), it is worthwhile to examine the Fourier transform of a pulse of amplitude A and duration T.

$$\text{let } v(t) = \begin{cases} A; & -T/2 < t < T/2 \\ 0; & \text{elsewhere} \end{cases}$$

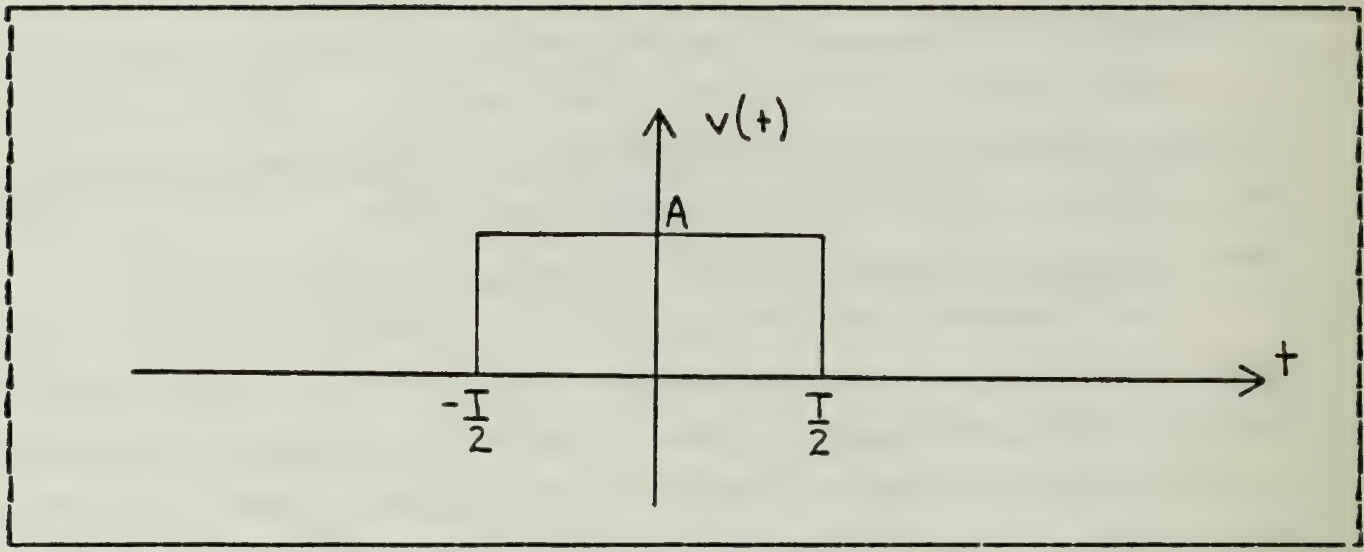



**Figure 3.1 Square Pulse**

Figure 3.1 is a representation of v(t) or the voltage expressed in the time domain. The position of v(t) on the t-axis was chosen for convenience of integration but could have been situated anywhere on the time line.

$$\mathcal{F}[v(t)] = V(w) = \int_{-\infty}^{\infty} v(t) e^{**}-jwt \, dt \qquad \text{(eqn 3.6)}$$

$$V(w) = \int_{-\infty}^{-T/2} e^{**}-jwt \, dt + \int_{-T/2}^{T/2} A \, e^{**}-jwt \, dt + \int_{T/2}^{\infty} 0 \cdot e^{**}-jwt \, dt \qquad \text{(eqn 3.7)}$$

$$V(w) = \int_{-T/2}^{T/2} A \, e^{**}-jwt \, dt \qquad \text{(eqn 3.8)}$$

The integration above may be attacked either head on or by substituting cos(wt) - j sin(wt) for e**-jwt. The direct approach is illustrated due to the relative simplicity of the integrand.

$$V(w) = \int_{-T/2}^{T/2} A \, e^{**}-jwt \, dt \qquad \text{(eqn 3.9)}$$

$$V(w) = -A/jw[e^{**}-jwt] \Big|_{-T/2}^{T/2} \qquad \text{(eqn 3.10)}$$

V(w) = -A/jw[e**-jwT/2 - e**jwT/2] (eqn 3.11)

V(w) = A/jw[e**jwT/2 - e**-jwT/2] (eqn 3.12)

By substituting 2pi f = w, the following result is obtained:

V(f) = A/j2 pi f[e**j2 pi fT/2 - e**-j2 pi fT/2] (eqn 3.13)

V(f) = A/j2 pi f[e**j pi fT - e**-j pi fT]

Using Euler's formula, i.e., sin θ = (e**jθ-e**-jθ)/2j:

V(f) = 2jA/j2 pi f[(e**j pi fT - e**-j pi fT)/2j] (eqn 3.14)

V(f) = A/pi f[sin(pi fT)]

Knowing that sin x / x = sinc x

V(f) = AT/pi fT[sin(pi fT)] (eqn 3.15)

V(f) = AT sinc(pi fT)

The sinc function is common in digital electronics and plots as the product of sin(pi fT) and 1/(pi fT) as in Figure 3.2. In Figure 3.2, V(f) in the frequency domain is equivalent to v(t) in the time domain. Note that as the pulse, T, gets longer, 1/T gets smaller or the first zero crossing of the sinc function occurs at a lower and lower frequency.

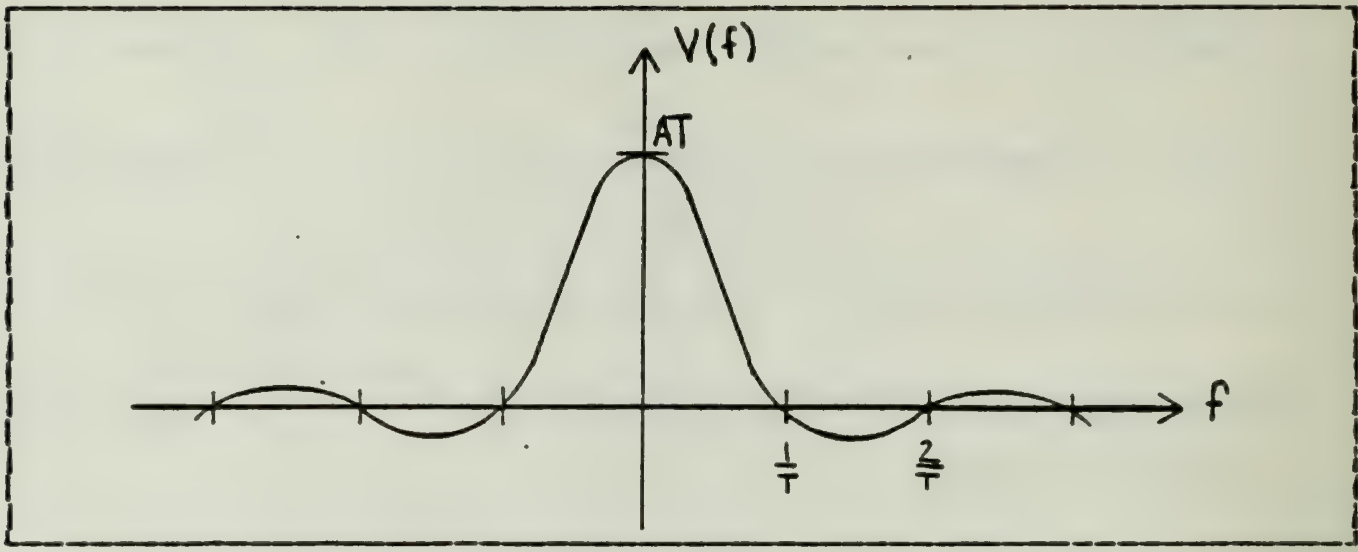



**Figure 3.2 Plot of Sinc Function**

1. Amplitude and Phase Spectrum

Recall that V(w) can be represented as in equation 3.16. The cosine term is the real part while the sine term is the imaginary part. By referencing Figure 3.3, a brief review of the complex plane is accomplished and its relationship to the amplitude and phase spectrum of a given voltage is represented. See equations 3.17 through 3.20.

$$V(w) = \underbrace{\int_{-\infty}^{\infty} v(t)\cos(wt)}_{\text{real}} - j\underbrace{\int_{-\infty}^{\infty} v(t)\sin(wt)}_{\text{imaginary}} \qquad \text{(eqn 3.16)}$$

$|V(f)|$ is called the amplitude spectrum of the given voltage. The amplitude spectrum can also be calculated by using the complex conjugate of the Fourier transform and is always positive as shown in equation 3.21. The phase spectrum, θ, of the function in question is represented by the arctan[imaginary/real] and is illustrated in Figure 3.5.

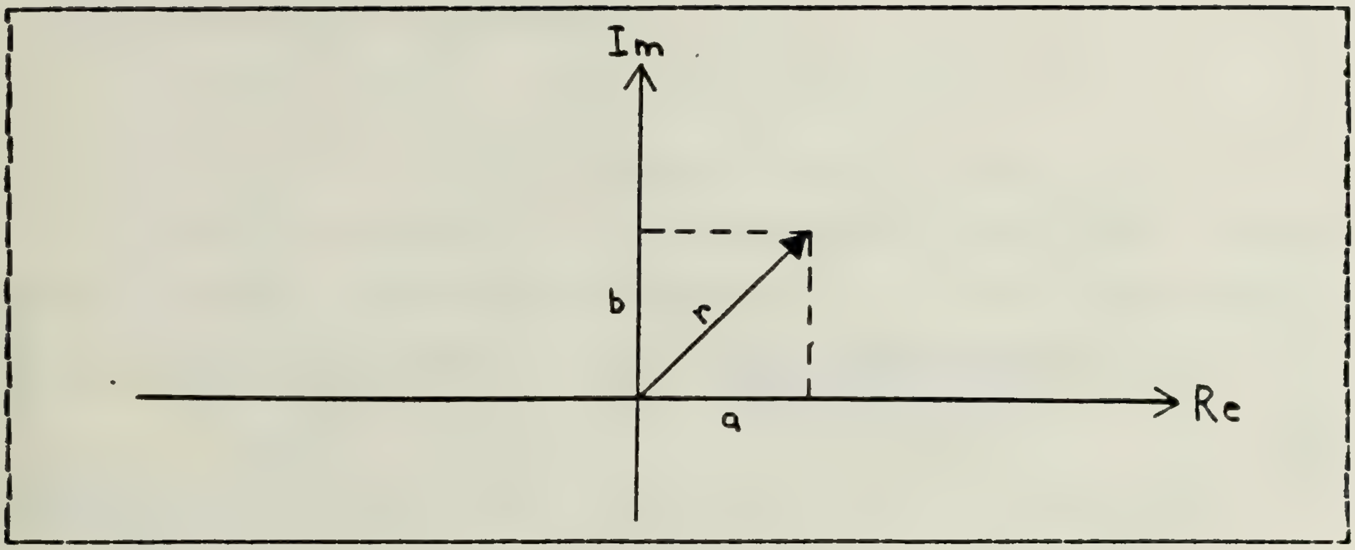



Figure 3.3 Phasor Diagram

$$r = [a^2 + b^2] \quad \text{(eqn 3.17)}$$

$$a + jb = [a^2 + b^2]^{**.5}\ e^{**j\theta} \quad \text{(eqn 3.18)}$$

$$V(f) = [\text{real}^2 + \text{imaginary}^2]^{**.5}\ e^{**j\theta} \quad \text{(eqn 3.19)}$$

$$|V(f)| = [\text{real}^2 + \text{imaginary}^2]^{**.5} \quad \text{(eqn 3.20)}$$

since $|e^{**j\theta}| = 1$

$$|V(f)| = [V(f) \bullet V^*(f)]^{**.5} \quad \text{(eqn 3.21)}$$

2. Properties of the Fourier Transform

Several properties of the Fourier transform are useful in the study of digital signals. They are represented here without proof and without a great deal of detail.

a. Linearity

if $v1(t)$ <-> $V1(f)$ and

if $v2(t)$ <-> $V2(f)$ then

$\mathcal{F}[av1(t) + bv2(t)] = aV1(f) + bV2(f)$

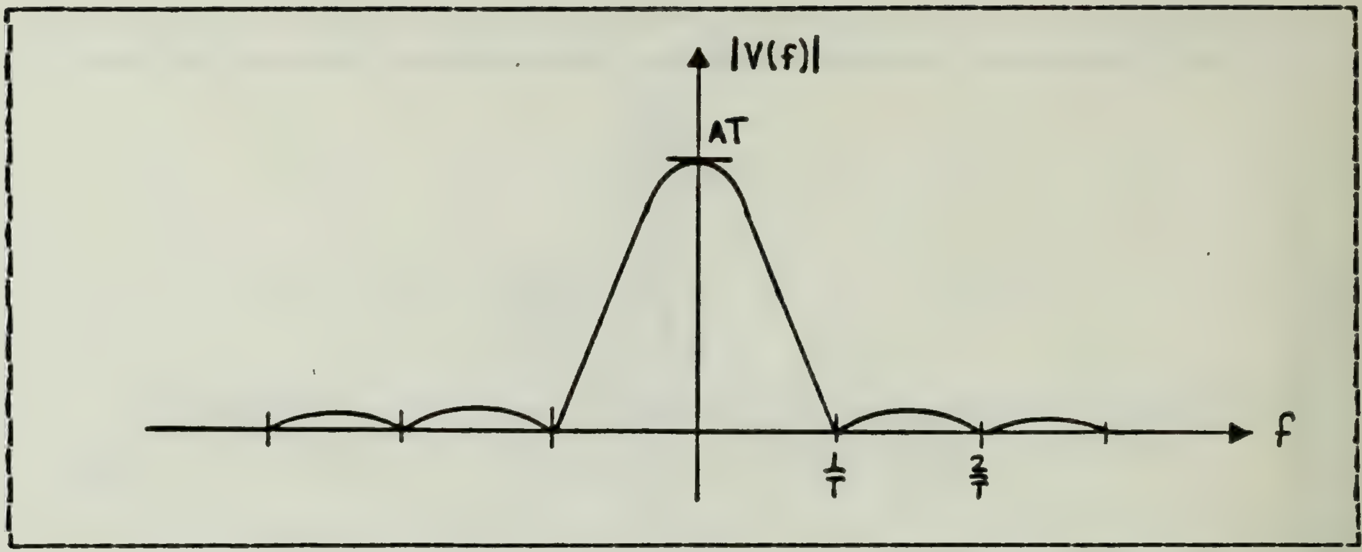



Figure 3.4 Amplitude Spectrum of Square Wave

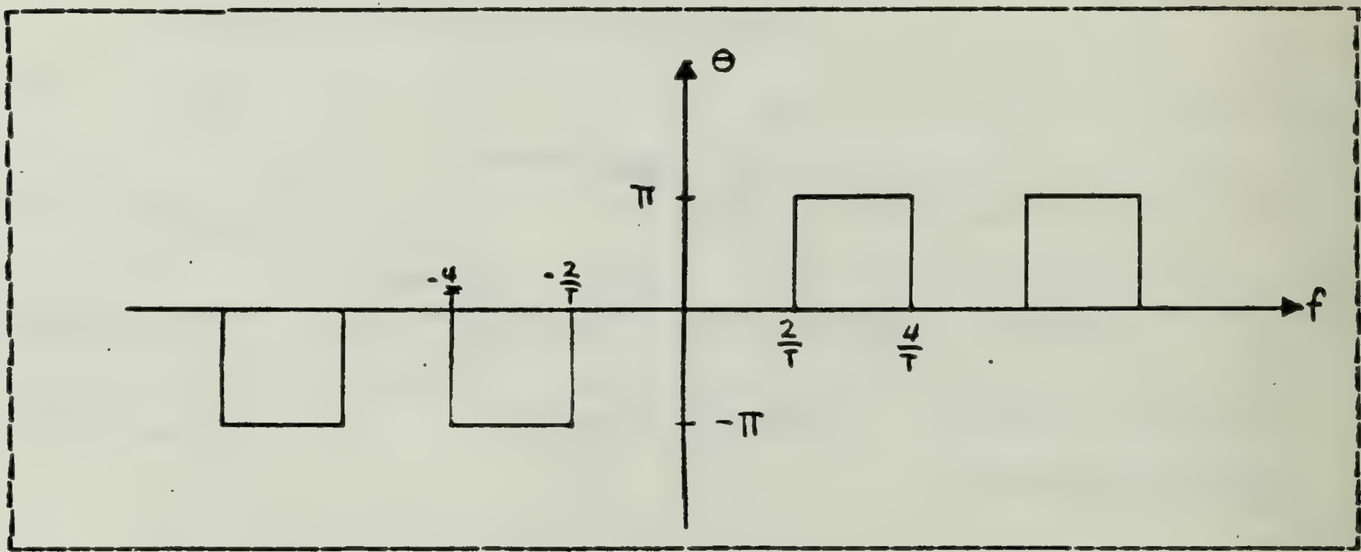



Figure 3.5 Phase Spectrum of Square Wave

b. Time-delay

$\mathcal{F}$[v(t-to)] = V(f) e**-jwto

Note that the amplitude spectrum of the delayed version is the same as the amplitude spectrum of the undelayed version. It is comforting to note that the Fourier transform of a pulse is the same tomorrow as it is today.

c. Scale change

$\mathcal{F}$[v(at)] = 1/|a| V(f/a)

d. Frequency translation

v(t)cos(2 pi fc t) <-> 1/2[V(f+fc) + V(f-fc)]

v(t) is any voltage

cos(2 pi fc t) is a carrier wave of frequency fc

Since understanding of this very important property of the Fourier transform is essential to the understanding of digital signal modulation it is expanded slightly here.

As we have seen, the Fourier representation of a voltage pulse of amplitude A and duration T is the sinc function of amplitude AT as illustrated in Figure 3.6.

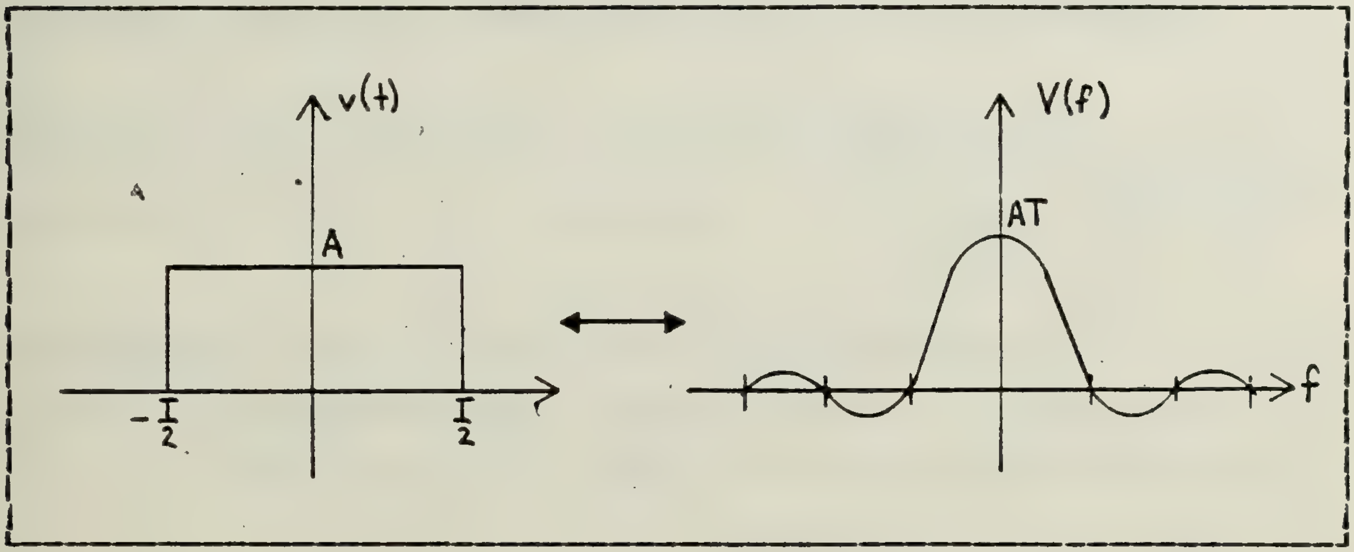



Figure 3.6 Square Wave in Time/Sinc Function in Frequency

The translation is the product of v(t) and in this case cos(2 pi fc t). In the time domain this product only exists in the interval between -T/2 and T/2 as in Figure 3.7. The significance of this translation and its relationship to the bandwidth of the voltage will be discussed further in the section dealing with bandwidth.

e. Differentiation

d v(t)/dt <-> jw V(f)

A differentiator could be used as a clock for timing but would never be used in the presence of noise since the

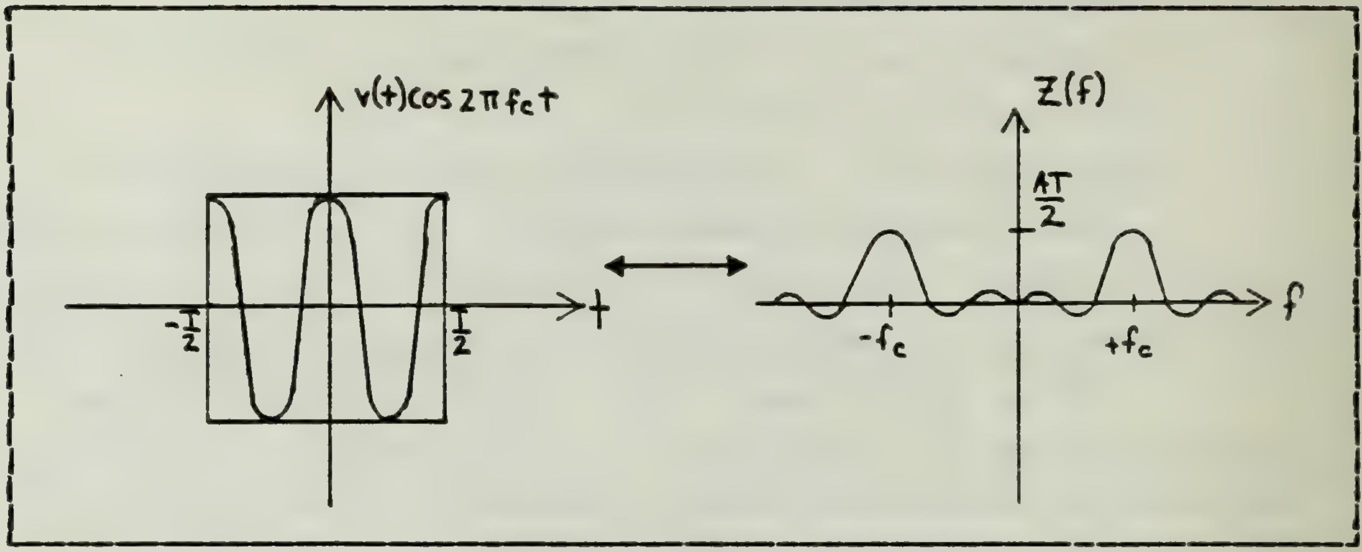



**Figure 3.7 Multiplication and Translation Diagrams**

result is exaggerated in the presence of high frequencies because w = 2 pi f.

f. Integration

v(t) dt <-> 1/w V(f)

An integrator could be used to reduce the effects of noise since w = 2 pi f is in the denominator tending to deemphasize the presence of high frequency noise.

## B. THE SAMPLING THEOREM

Essential to the understanding of digital communications is the sampling theorem which was first introduced by Nyquist in 1928 [Ref. 9], and later by Shannon in 1948 [Ref. 10]. The sampling theorem states that any voltage can be uniquely represented by appropriately spaced sample values of the original voltage. More correctly stated, the sampling theorem places limits on the accuracy with which a signal can be represented.

The implication is that an analog signal can be represented digitally or by a set of numbers, i.e., samples. A description of the sampling theorem follows.

Given any analog voltage, v(t) as in Figure 3.8, the sampling theorem says that the entire analog signal is not required to accurately represent the voltage but only samples of it, call them vs(t). Figure 3.9 shows samples of a representative analog voltage. Samples can be taken of v(t) at every T seconds for a period of seconds. This can be accomplished by the use of a voltage clock, call it vc(t). The sampling can be viewed graphically as a block diagram representing an analog voltage multiplier as shown in Figure 3.10. To be of further use in the understanding of digital communications we are interested in a frequency description of the sample voltage, vs(t). Note that the system which describes the obtaining of vs(t) involves a voltage multiplication. It was demonstrated in the proceeding section that voltage multiplication amounted to frequency translation, a property of the Fourier transform.

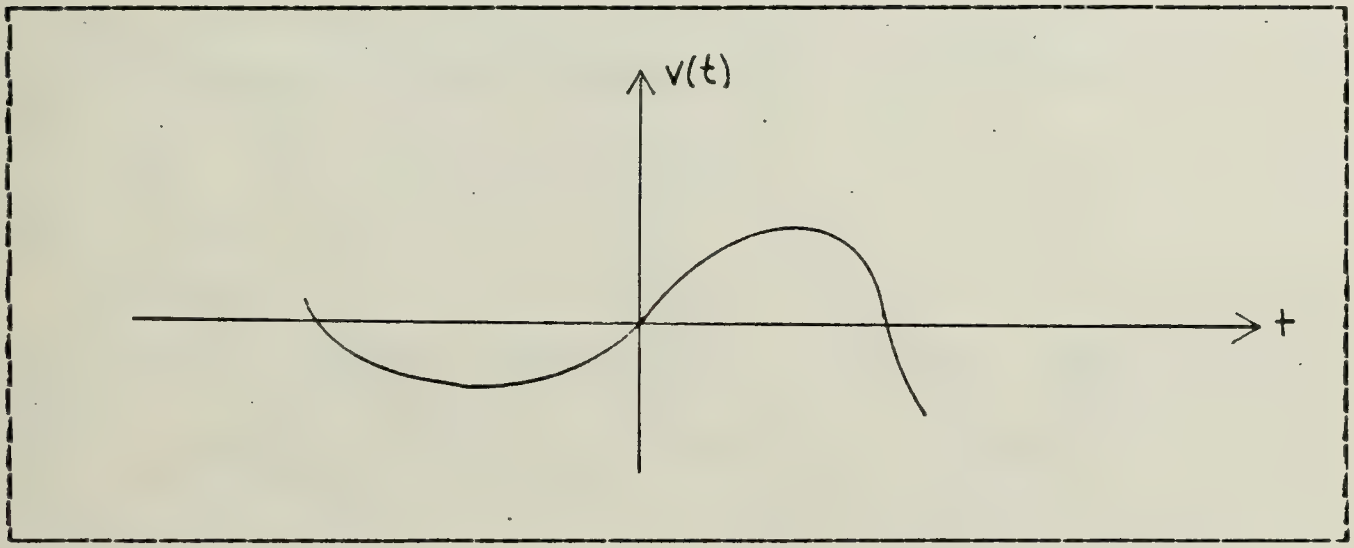



Figure 3.8 Representative Analog Voltage

First it is necessary to find the frequency description of the clock, vc(t), Let vc(t) be a periodic square wave of height 1 and duration d as in Figure 3.11. Since vc(t) is periodic, it can be shown that the Fourier series representing vc(t) is given by equation 3.22.

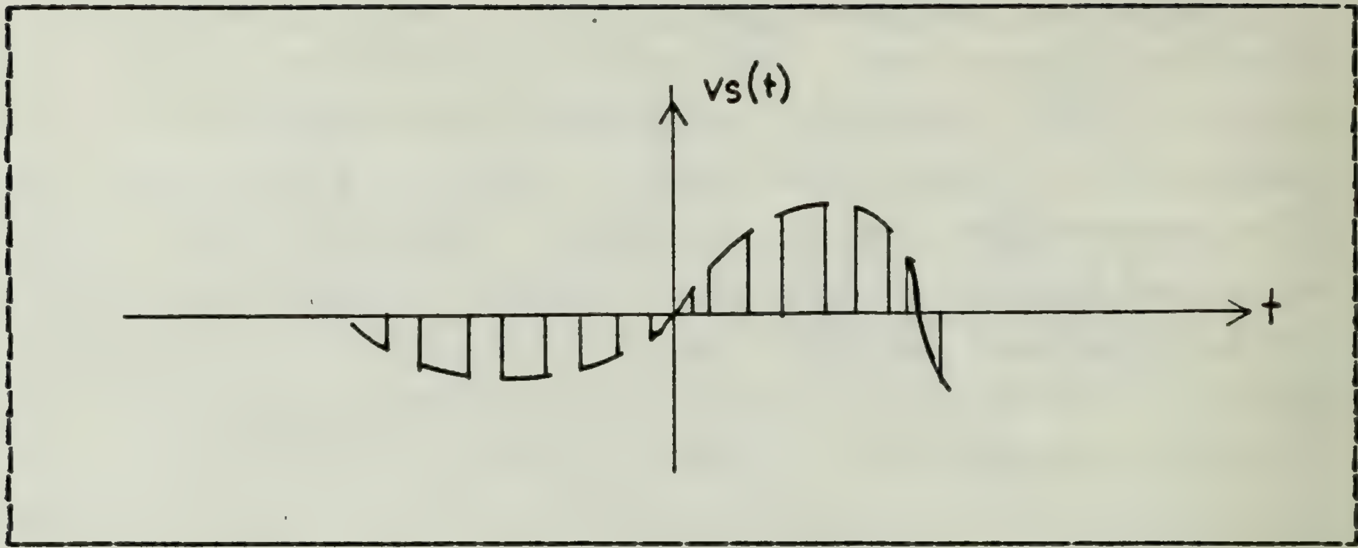



**Figure 3.9 Samples of a Representative Voltage**

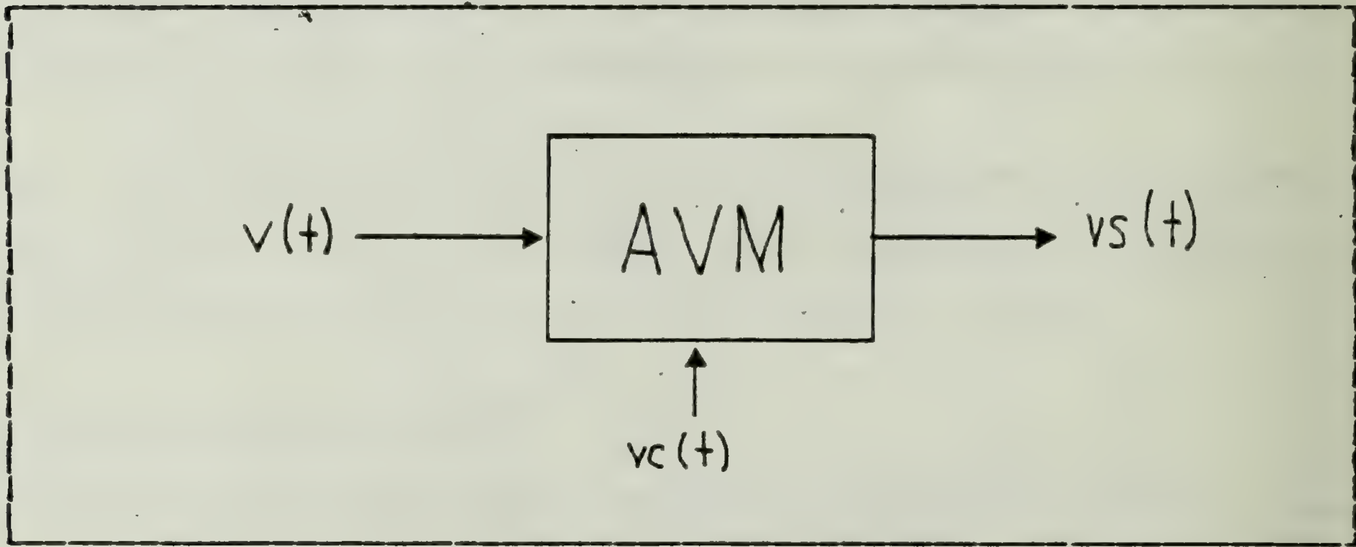



**Figure 3.10 Analog Voltage Multiplier**

Again it is noted that vs(t), the sample voltage, is the product of v(t), the original analog voltage, times vc(t), the clock voltage. In other words vs(t) = v(t) times a series of cosine terms.

$$vc(t) = ao + \sum_{n=1}^{\infty} an \cos(2\ pi\ n\ fc\ t) \qquad \text{(eqn 3.22)}$$

ao and an are left unevaluated

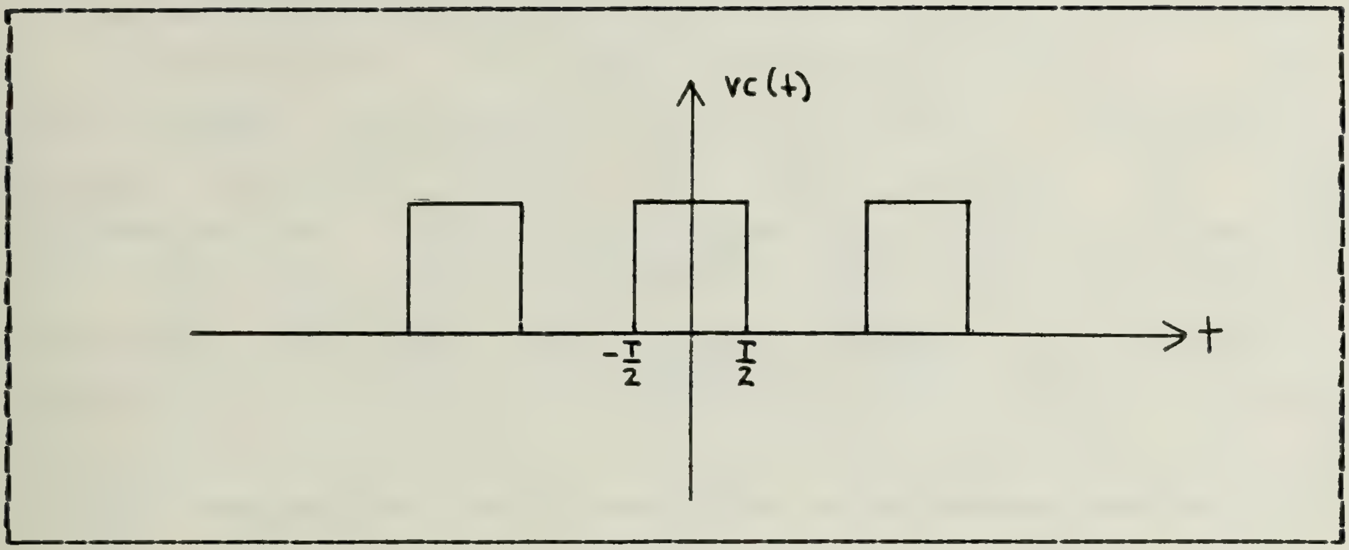



**Figure 3.11 A Representative Voltage Clock**

$$vs(t) = v(t)[ao + \sum_{n=1}^{\infty} an \cos(2 \text{ pi } n \text{ fc } t)] \qquad \text{(eqn 3.23)}$$

Now, assuming any form of the Fourier transform of v(t), as in Figure 3.12 and since vs(t) is a product and by utilizing the frequency translation property of the Fourier transform, the following is obtained as a representation of the frequency spectrum of vs(t). See Figure 3.13.

Remember that the curve highlighted in the box in Figure 3.13 is the Fourier represention of v(t), the original signal. This original signal can now be recovered by filtering with an appropriate low pass filter of bandwidth equal to or greater than B. This low pass filter would only permit the reception of that portion of the signal which represents the original voltage.

The only question left to resolve is how often to take a sample. Again referring to the diagram in Figure 3.13, it is noted that in order to prevent any overlap of successive translations (aliasing)

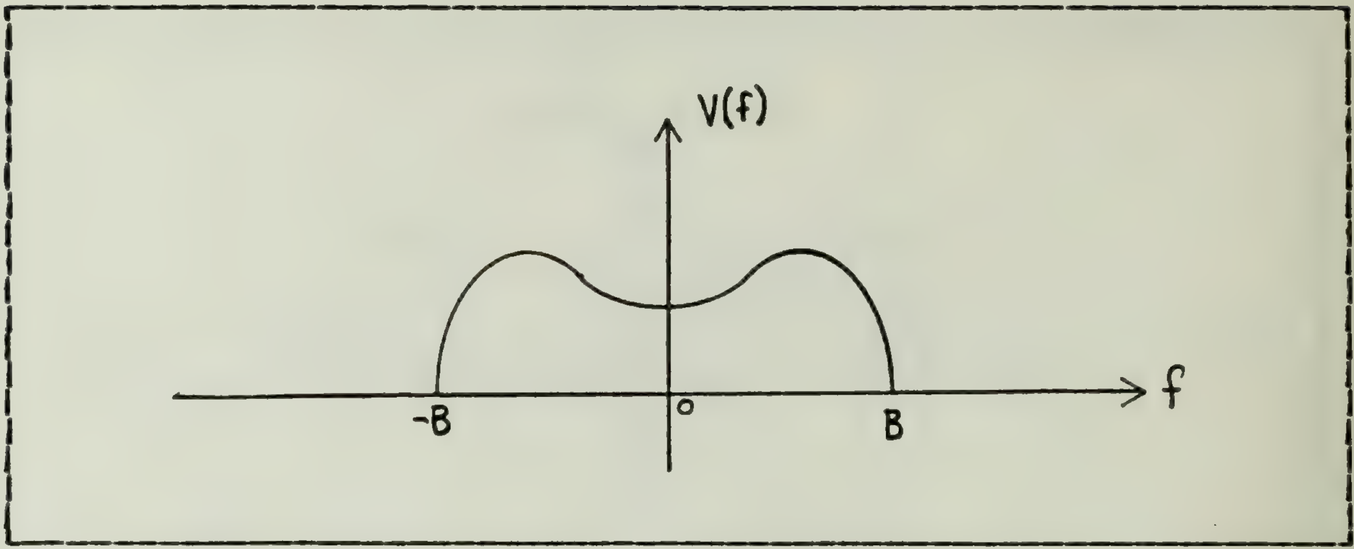



Figure 3.12 Fourier Transform of v(t)

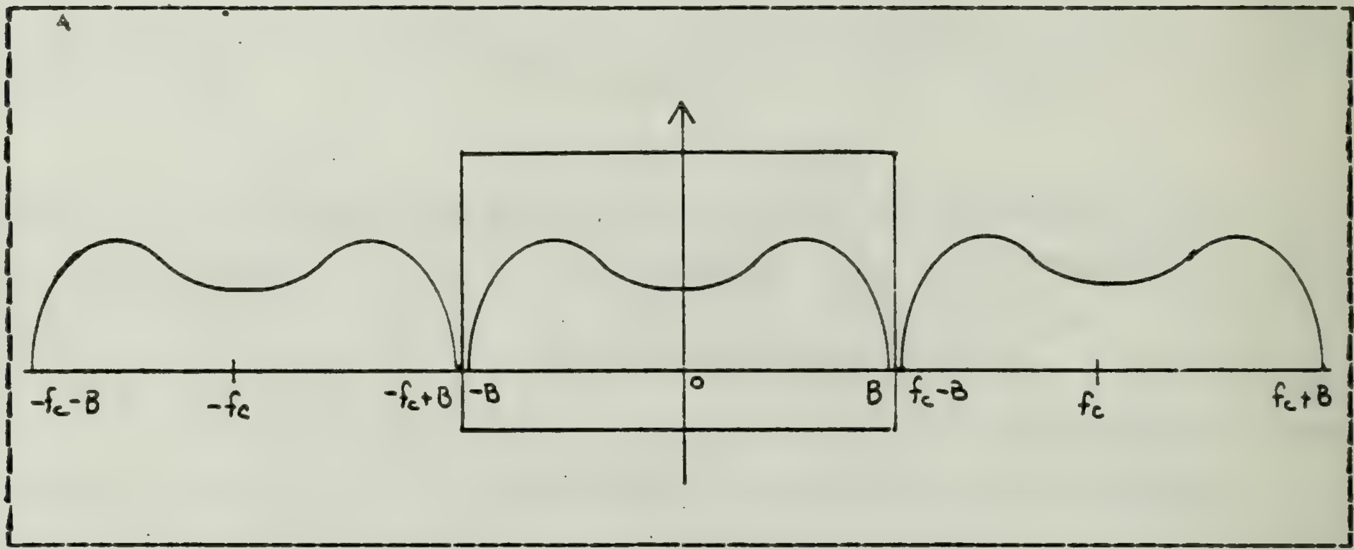



Figure 3.13 Fourier Transform of vs(t)

fo - B >> B

fo >> 2B

or the frequency of the clock, fc (the sampling rate = 1/T) must be strictly greater than 2 times the largest significant Fourier frequency component present. Since B will vary with different v(t), the sampling rate necessary to uniquely recover that v(t) also varies.

In summary on the sampling theorem, it can be said that any analog signal can be represented as certain sample values spaced the appropriate distance apart. The sampling theorem places limits on the accuracy of this representation. These sample values can then be transmitted from one position to another using analog to digital conversion and any one of a variety of modulation techniques. It has been shown that the Fourier frequency representation of the sample is not equivalent to the Fourier frequency representation of the source voltage; however, the source voltage can be recovered at the receiving end by filtering. A block representation of the system is shown in Figure 3.14.

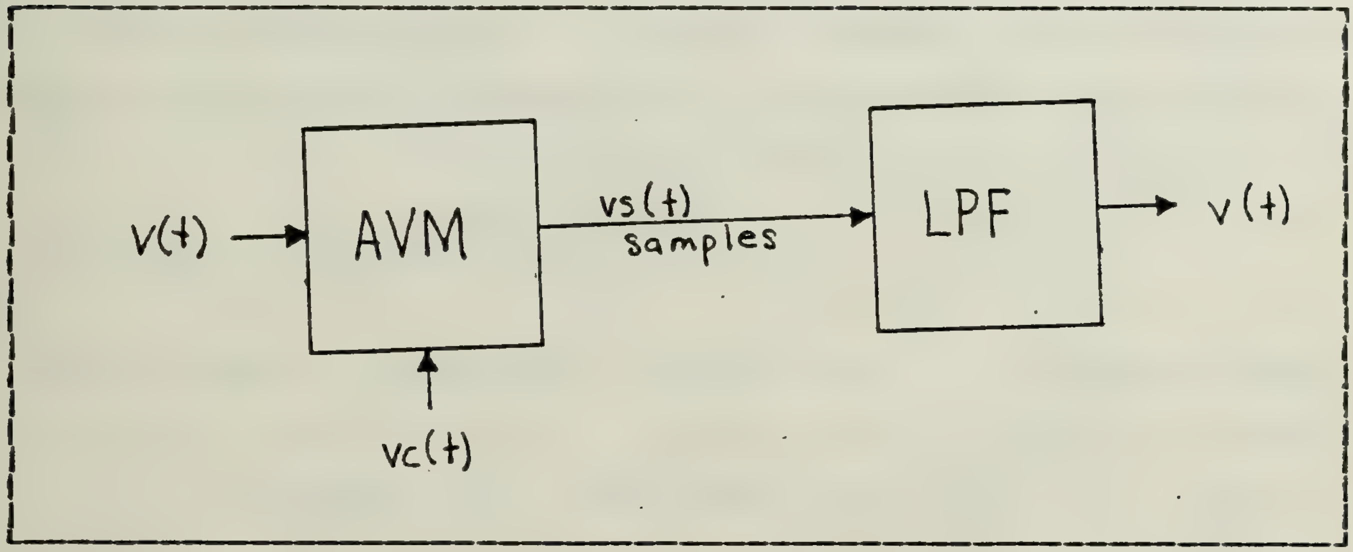



Figure 3.14 Analog Voltage Multiplier and Low Pass Filter

## C. THE AUTOCORRELATION FUNCTION

The above concept certainly seems simple enough. If a signal is present and it is desired to transmit it from point A to point B, all that has to be done is to take appropriately spaced samples of the signal, encode them, somehow transmit them to point B, filter the received signal and the uniqueness of the original voltage has been reproduced. Unfortunately it isn't quite that easy. The reason it is not, is due to the presence of noise. The sources of noise will not be discussed here, however, the effects of noise in general terms and how a faint signal can be recovered in the presence of noise will be discussed.

The time varying descriptor of voltage and the frequency varying descriptor of voltage have been introduced. Both are precise mathematical descriptions of something (voltage) that is deterministic. That is, it can be described in mathematical or graphical terms during any period of time. Such is not the case for noise corrupted voltages which are entirely random. Therefore other descriptors of voltages must be employed in the presence of noise. They are partial descriptors of voltage since a random signal cannot be described with precision. These partial descriptors are the

1. Autocorrelation function and the
2. Probability density function (p.d.f.)

It is important here to note the difference between the source voltage at the receiver and the sample voltage. The source voltage at the receiver, vsr(t), is the digitally converted sample voltage, vs(t), modulated onto a carrier by one of the digital modulation techniques yet to be discussed. Understanding the autocorrelation function is the basis for understanding how a decision is made that vsr(t) is present at the receiver in the presence of noise. It should be remembered that the exact form of the carrier is known at both the transmitter and receiver.

First of all, an additive noise model is assumed where the signal at the receiver, vr(t), is equal to the signal which is being transmitted, vsr(t), (i.e., the digital samples of v(t) modulated onto the carrier), plus random noise, vn(t)

$$vr(t) = vsr(t) + vn(t) \qquad \text{(eqn 3.24)}$$

vr(t) = voltage at the receiver

vsr(t) = signal voltage at the receiver

vn(t) = noise voltage at the receiver

If it is realized that most receivers operate on the principal of detection of DC voltage, the problem becomes one of rectification of the received signal and determination if the DC voltage, characteristic of the transmitted signal is present.

A common type of rectification of an analog voltage involves squaring the input waveform. If the given signal at the receiver is vr(t) as in equation 3.24 above, squaring the waveform introduces the square of not only the desired signal but also the square of the noise term. If instead of squaring vr(t) and introducing or at least not eliminating the noise, vr(t) is multiplied by vsr(t), the results in equation 3.25 are obtained.

$$vr(t) \cdot vsr(t) = vsr**2(t) + vsr(t)\,vn(t) \qquad \text{(eqn 3.25)}$$

By averaging, all that remains is $\overline{vsr**2(t)}$ since the average of vsr(t)vn(t) is zero because $\overline{vn(t)}$ is random and the average of any random voltage is zero. The DC component of vr(t)•vsr(t) is $\overline{vsr**2(t)}$. Any remaining AC component can be removed by a low pass filter. Graphically in block diagram form this receiver looks like Figure 3.15 illustrated below. If vsr**2(t) > 0 then vsr(t) is present in the signal vr(t). If vr (t) is pure noise or some other signal is present exclusively, then the output of the receiver will be 0.

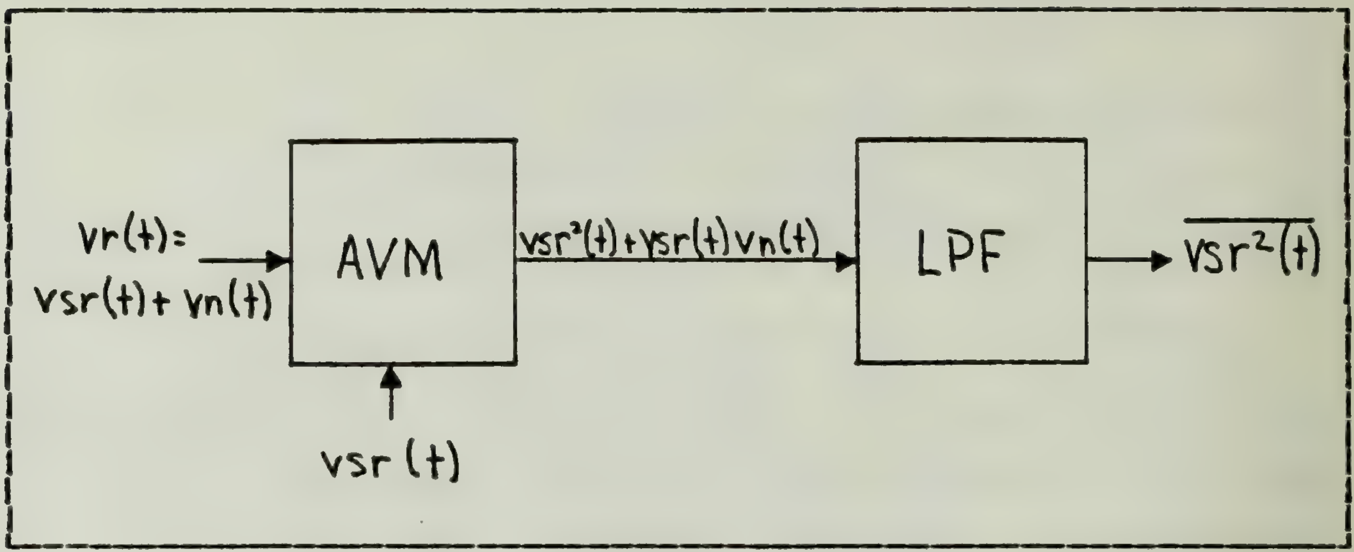



Figure 3.15   Rectification through an AVM and LPF

In practice, however, the exact waveform of vsr(t) may not be known or it may be a time delayed or distorted version of the original signal when it arrives at the receiver. If the distortion or delay is significant enough there will be little agreement or "correlation" in the receiver.

The solution is to multiply the received signal by a series of time delayed approximations of the transmitted signal. The signal is present whenever the output of the series of voltage multipliers or correlators exceeds a certain threshold approaching vsr**2(t). The concept is illustrated in Figure 3.16.

For the concept illustrated in Figure 3.16 to work, the average value of vr(t)•vsr(t-p) must have a DC component.

$$vr(t) \bullet vsr(t-p) = vsr(t)\,vsr(t-p) + vn(t)\,vsr(t-p) \qquad \text{(eqn 3.26)}$$

$$\overline{vr(t) \bullet vsr(t-p)} = \overline{vsr(t)\,vsr(t-p)} + \overline{vn(t)\,vsr(t-p)} \qquad \text{(eqn 3.27)}$$

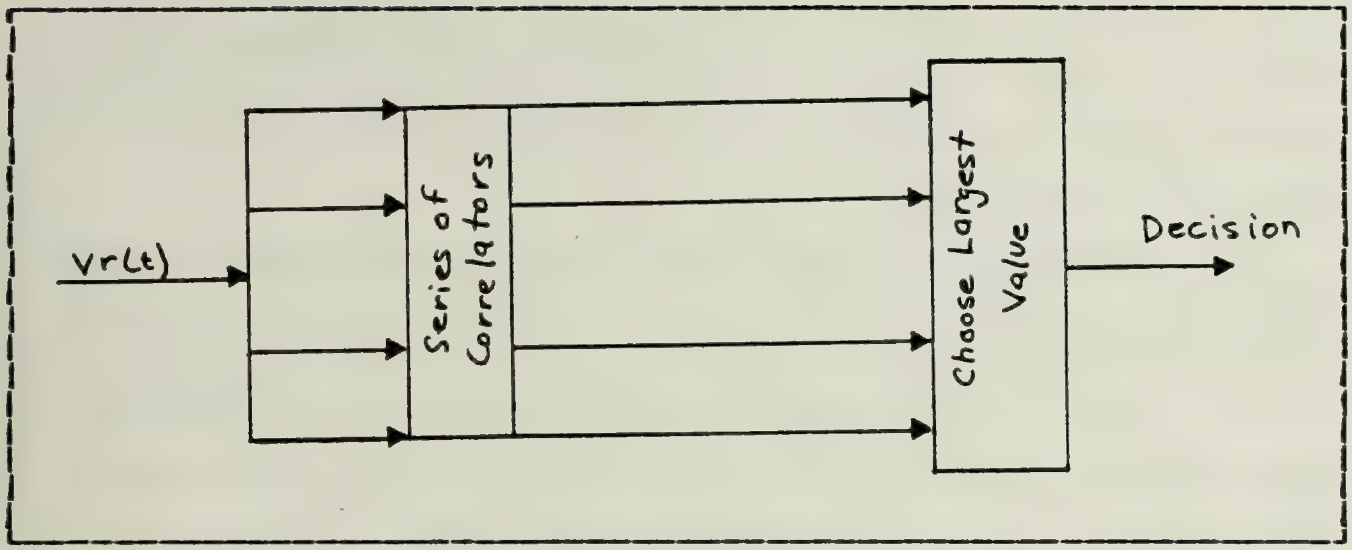



**Figure 3.16 Correlation Device**

But the average of vn(t)vsr(t-p) = 0 since $\overline{vn(t)}$ is random and the results of equation 3.28 are obtained.

$$\overline{vr(t) \cdot vsr(t-p)} = \overline{vsr(t)\,vsr(t-p)} \qquad \text{(eqn 3.28)}$$

What is of interest is the average value of vsr(t)vsr(t-p). The autocorrelation function (ACF) will be defined as that average value of vsr(t)vsr(t-p). The mathematical representation of the autocorrelation function for a continuous voltage is represented in equation 3.29 while the autocorrelation function for a pulse voltage is presented as equation 3.30.

$$ACF = Rvv(p) = 1/2T \int_{-T}^{T} v(t)\,v(t-p)\,dt \qquad \text{(eqn 3.29)}$$

$$ACF = Cvv(p) = \int_{-\infty}^{\infty} v(t)\,v(t-p)\,dt \qquad \text{(eqn 3.30)}$$

The autocorrelation function is a measure of the degree to which two identical signals which are corrupted in some manner are alike. The ACF of two signals which are identical, not distorted in any way and occurring at exactly the same time, i.e., $p = 0$ has a maximum value. On the other hand there will be very little correlation between two identical time variant signals when the time difference between them is great. In this case the autocorrelation function is near zero.

A similar concept is used within the context of signal comparison. This concept is the crosscorrelation which is a measure of the degree to which 2 different signals are alike. When the crosscorrelation between voltage vsr and random noise voltage vn is 0 there is no relationship or correlation.

Again let us assume that the signal at the receiver is a noise disrupted version of the signal originally transmitted.

$$vr(t) = vsr(t) + vn(t) \qquad \text{(eqn 3.31)}$$

In the receiver , the time delayed version of the signal is applied to the incoming signal and the average is formed as before and shown in Figure 3.17.

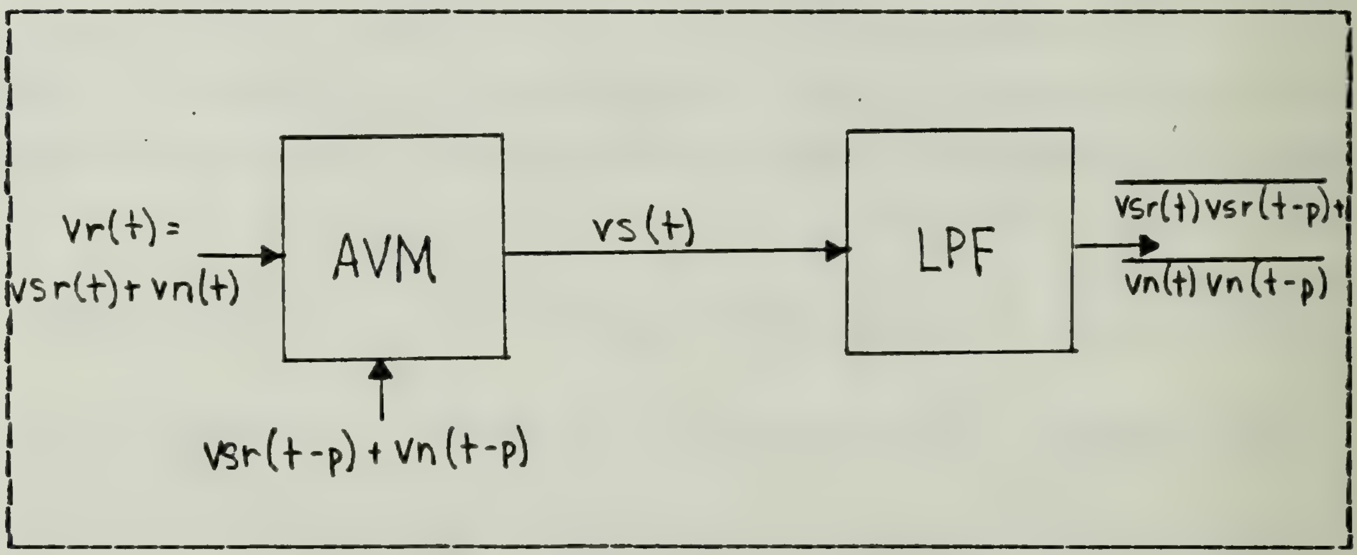



Figure 3.17 Voltage Correlator

$$\overline{[\,vsr(t) + vn(t)\,]} \cdot \overline{[\,vsr(t-p) + vn(t-p)\,]} = \qquad \text{(eqn 3.32)}$$

$$\overline{vsr(t)\,vsr(t-p)} + \overline{vn(t)\,vsr(t-p)} +$$

$$\overline{vsr(t)\,vn(t-p)} + \overline{vn(t)\,vn(t-p)} =$$

$$R_{vsr\ vsr}(p) + R_{vn\ vsr}(p) + R_{vsr\ vn}(p) + R_{vn\ vn}(p)$$

$$R_{vn\ vsr}(p) = 0$$

$$R_{vsr\ vn}(p) = 0; \text{ because average } vn = 0$$

Therefore the result is simply $R_{vsr} = vsr(p) + R_{vn}\ vn(p)$ or the autocorrelation function of the desired signal plus the autocorrelation function of the noise. Assuming the shape of the autocorrelation function of both components (both the original signal and the noise which vary with p, time) is known, what is done in practice is to vary the value of p until the ratio of $R_{vsr}\ vsr(p)$ / $R_{vn}\ vn(p)$ is a maximum or the autocorrelation function of the signal is a maximum while the autocorrelation of the noise is minimum and the signal is recovered.

## D. THE MATCHED FILTER

The matched filter is a linear filter which has the characteristic response desired for optimum reception of the desired signal. In other words, we desire the response of the system to the linear filter to be in some way proportional to the autocorrelation function of the desired signal and in no way related to the autocorrelation function of the noise. This system could be illustrated as in Figure 3.18.

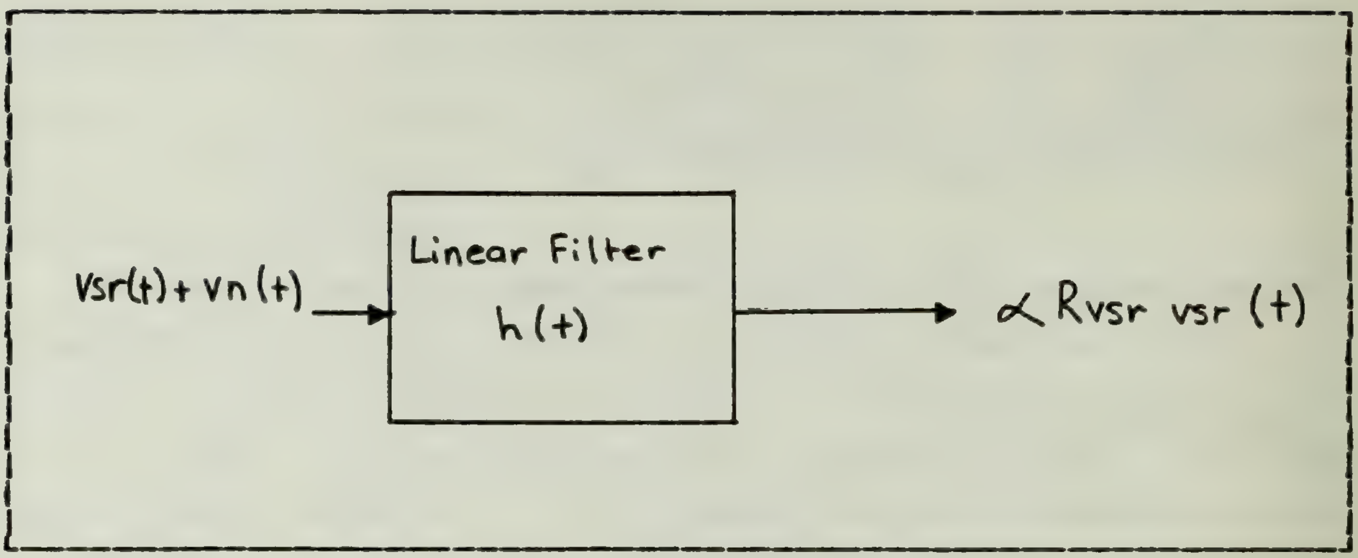



**Figure 3.18 Matched Filter Block Representation**

The question is what should the makeup or h(t), (i.e., the response of the linear filter) be? The output of a linear system is a convolution so the output of the linear filter, h(t) to an input vsr(t) would be vsr(t) * h(t), where * denotes convolution.

In the previous section it was stated that the autocorrelation function, Rvsr vsr(p), a function of p, is given by

$$R_{vsr\ vsr}(p) = \int_{-\infty}^{\infty} vsr(t)\,vsr(t-p)\,dt \qquad \text{(eqn 3.33)}$$

By simply changing variables, equation 3.33 can be rewritten as a function of time, t.

$$R_{vsr\ vsr}(t) = \int_{-\infty}^{\infty} vsr(p)\,vsr(p-t)\,dp \qquad \text{(eqn 3.34)}$$

From convolution theory it is known that the response of a linear filter as above, vsr(t) * h(t) can be expressed mathematically as in equation 3.35.

$$vsr(t) * h(t) = \int_{-\infty}^{\infty} vsr(p)\,h(t-p)\,dp \qquad \text{(eqn 3.35)}$$

As stated earlier, the desired output of the linear filter is the autocorrelation function Rvsr vsr(t). Note the similarities between the last two equations. If in equation 3.35 a simple substitution of h(t) = vsr(-t) is performed, the desired results are obtained.

$$vsr(t) * h(t) = \int_{-\infty}^{\infty} vsr(p)\, vsr(p-t)\, dp \qquad \text{(eqn 3.36)}$$

$$= R_{vsr\ vsr}(t)$$

Therefore, for every signal that is desired to be recovered, the matched filter will do the job very nicely if the response of that filter, h(t) is equal to the inverse of the signal that is desired to be detected.

# IV. ANALOG TO DIGITAL CONVERSION

Key to the understanding of digital communications is how the analog information is converted to digital information. As illustrated earlier, each analog voltage or signal can be represented by appropriately spaced sample values. These sample values are still analog in that they can take on any value in the range of the original analog signal. What is desired is to change this infinite set of decimal numbers to a finite set of decimal numbers. This is called analog to digital conversion. Common types of A to D conversion include Pulse Code Modulation (PCM), Differential Pulse Code Modulation (DPCM), Delta Modulation (DM), Pulse Amplitude Modulation (PAM), Pulse Duration Modulation (PDM), and Pulse Position Modulation (PPM). PCM, DPCM, and DM will be examined in this chapter because of their widespread usage and easy application to digital technology.

## A. PULSE CODE MODULATION (PCM)

PCM is the most widely used A to D conversion technique. It involves assigning the sample value obtained in sampling to one of several quantization levels within the range of the voltage sampled and then representing those quantization levels by various binary code words.

For example, the highest significant frequency component in human speech is 3300 Hz. In order to capture accurately the signal produced in speech, one would require a sampling rate greater than or equal to 2B or 2(3300) = 6600 samples/sec. The telephone company uses 8000 samples/sec.

$B = 3300$ Hz

$f \geq 2B = 6600$ samples/sec

use $f = 8000$ samples/sec to ensure no aliasing

If a signal is sampled at the rate of 8000 samples/sec, then in the transmission of those samples, assuming the samples are transmitted immediately and not delayed, the transmission rate must be

Ts = 1/f

Ts = 1/8000 samples/sec

Ts = 125 microsec/sample

The number of quantization levels employed to represent the various analog sample values is arbitrary. The greater the number of levels, the more accurate the reconstruction of the original signal but the more rapid the data transmission rate must be(in all cases there will be some error present after recovery). Consider again the example of voice as illustrated in Figure 4.1.

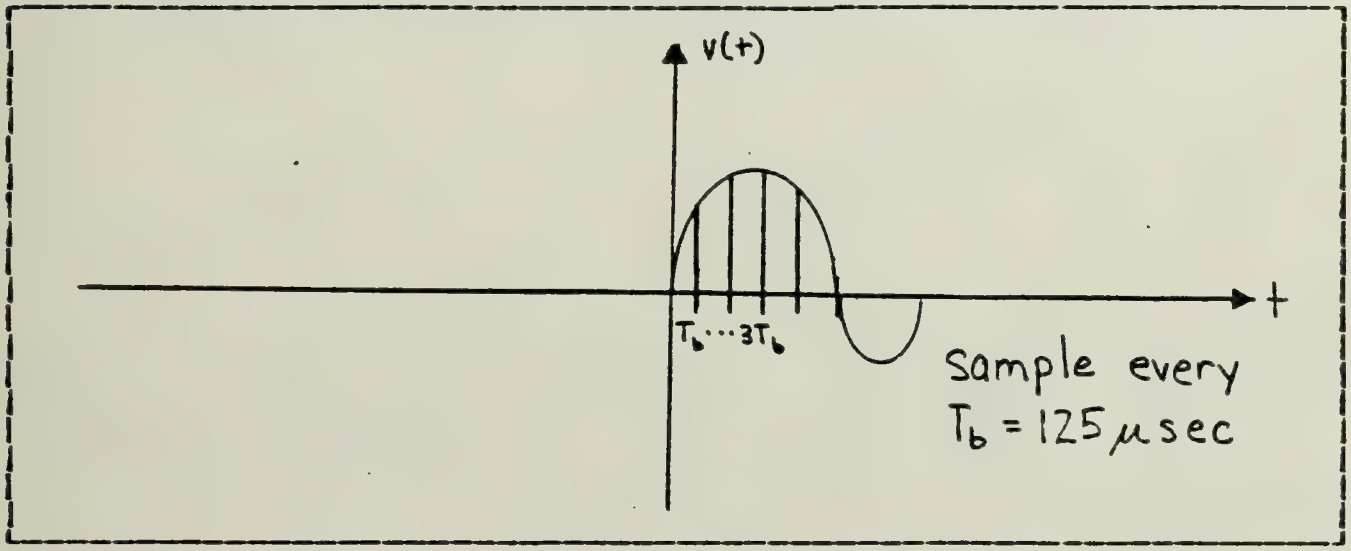



**Figure 4.1 Samples of Voltage**

Assume it is desired to represent each sample value (decimal number) with an 8 bit binary code word. Then the number of quantization levels is given by equation 4.1.

The spacing between quantization levels is the range of voltages to be represented divided by the number of quantization levels. The analog to digital conversion takes

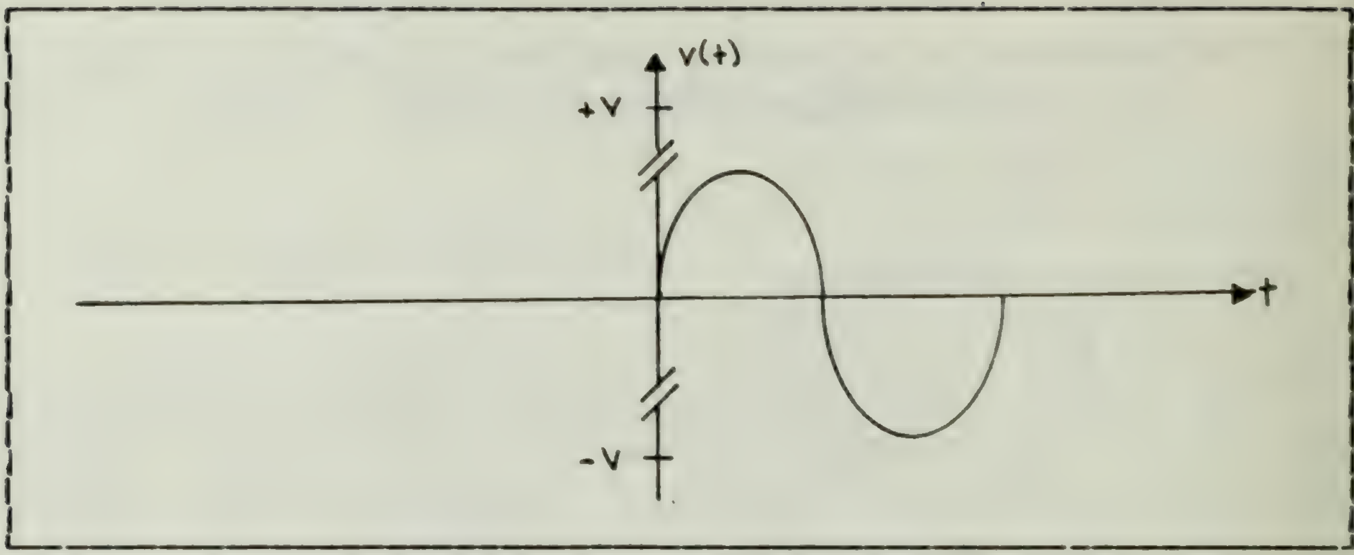



**Figure 4.2 Analog Nature of Samples**

#quantization levels = 2**n (eqn 4.1)

for n = 8 bits/binary code word
#quantization levels = 2**8
= 256

place by determining into which quantization level the respective sample values fall and assigning that sample value the binary number associated with that quantization level.

Now each sample value is assigned to an eight bit binary code word. For voice transmission it has been shown that the minimum transmission rate was Ts = 125 microsec/sample. The bit rate is then simply equal to the number of bits per sample times the inverse of the transmission rate as illustrated in equations 4.2 and 4.3.

If voice was to be time multiplexed onto a single channel, the bit rate for that channel would be the number of signals per channel times the bit rate for the most time intensive signal as illustrated in equation 4.4.

bit rate = (n bits/sample)(1 sample/T sec) (eqn 4.2)

bit rate = (8 bits/sample)(1 sample/125 microsec)

= 64 kbps

bit duration = 1/bit rate (eqn 4.3)

= 1/64 kbps = 15.6 microsec/bit

(#signals/channel)(bit rate/signal) = (eqn 4.4)

bit rate/channel

In other words, if 24 voice signals are to be multiplexed onto a single channel, the bit rate must be

bit rate/channel ≥ (eqn 4.5)

(24 signals/channel)(64 kbps/signal)

= 1536 Mbps

As can be seen, the data transmission rate increases significantly, necessitating better and better hardware and a wider and wider bandwidth. Advantages should be readily apparent for A to D conversion techniques or modulation techniques that reduce the bit rate required.

Recovery of the analog signal is by a process called Digital to Analog Conversion. All that will be said about D to A conversion is that it is the inverse process of A to D conversion and that a slight error is always introduced during the process due to the discrete nature of the quantization process during A to D and D to A conversion.

## B. DIFFERENTIAL PULSE CODE MODULATION (DPCM)

In DPCM, what is converted to binary numbers is not the quantization level (decimal number) which each sample value is represented by but the successive differences between quantization levels. The idea is that the range of maximum difference values will be smaller than the range of actual sample values. It is therefore possible to represent that range of delta values with fewer bits/binary word and therefore fewer quantization levels and ultimately a lower bit rate. [Ref. 11]

In actual practice, in a DPCM conversion technique, it is a statistical estimate of each successive sample value which is subtracted from the actual value that is converted to binary code words. The result is the same in that the range of amplitudes is reduced and therefore fewer bits/word are required to represent the sample thereby lowering the data transmission rate required to transmit the signal. [Ref. 11]

## C. DELTA MODULATION (DM)

Delta modulation is a form of DPCM where successive quantization levels of the output differ by only 1 bit. That is to say that any successive quantization level can be represented by varying only one bit of successive output binary code words. In other words a type of gray code is employed. This A to D technique is implemented through the use of a DM coder or linear delta modulator. This DM coder approximates a given input signal with a series of linear segments of uniform slope. A comparison is made between the value of this approximation and the input signal at each sample increment. The sign of this difference value is what is encoded and is used to increment the DM coder in the direction of the input signal. By using the differential sign value of the input signal and the incremented approximation from the DM coder the linear approximation from the linear delta modulator is said to "track" the input signal. [Ref. 8] Slope overlcad of this type of modulation technique occurs when the slope of the incoming signal exceeds the ability of the DM system to follow the source at the sampling rate being utilized. [Ref. 11].

# V. DIGITAL SIGNAL MODULATION TECHNIQUES

Digital signal modulation techniques are the methods of encoding information for transmission utilizing digital technology. Factors affecting digital modulation include, but are not limited to, the physics of the method, hardware requirements, bandwidth considerations, power requirements, data transmission rates and error probabilities. It is these aspects which will be explored in the following sections.

## A. DIGITAL MODULATION FORMATS

Modulation is the technique by which the characteristics of one waveform (called the carrier) are varied or modified by the characteristics of another (called the source). The carrier waveform of interest is the sinusoid. It should be obvious that the attributes of a particular sinusoid that differentiate it from every other sinusoid are its amplitude, phase and frequency. It follows that the characteristics of the carrier waveform to be varied by the source are its amplitude, phase, frequency or any combination of the three. This gives rise to the broad general formats of Amplitude Shift Keying (ASK), Phase Shift Keying (PSK) and Frequency Shift Keying (FSK). All other digital modulation techniques are variations of or combinations of these basic formats. Other factors involved in the description of the digital modulation technique being employed include the number of bits being encoded at one time, the employment of error correction techniques and the baseband (source) waveform.

When each bit of the baseband waveform is individually encoded by any of the techniques previously mentioned (ASK, PSK, FSK), the technique being utilized is referred to as binary encoding. When more than 1 bit of the source code is modulated onto the carrier at one time it is called block encoding. Block encoding allows for one of m = 2**k waveforms where k = number of bits.

This paper will not go deeply into signal detection or demodulation techniques, however, a basic understanding of what is involved is necessary and again further defines the digital modulation format being employed. In general terms, signal detection is referred to as either coherent or noncoherent detection of the transmitted signal. Coherent detection, perhaps the easiest to understand, is when all possible waveforms of the modulated carrier waveform are available at the receiver and the waveform at the receiver is in phase with the transmitted carrier. Noncoherent detection is involved when the receiver does not have knowledge of the phase of the transmitted information and one of a number of phase estimation techniques must be employed for signal recovery.

## B. HARDWARE

Although significant advances have been made in recent years to improve the quality of hardware associated with satellite communications, most components are not what would be considered "off the shelf items". The hardware components most commonly referred to with regard to digital communications include the sampler, encoder, modulator, multiplexer and transmitter. There are variations of the above hardware requirements necessary for certain types of digital formatting, however, those exceptions will be addressed separately when the individual modulation techniques are examined.

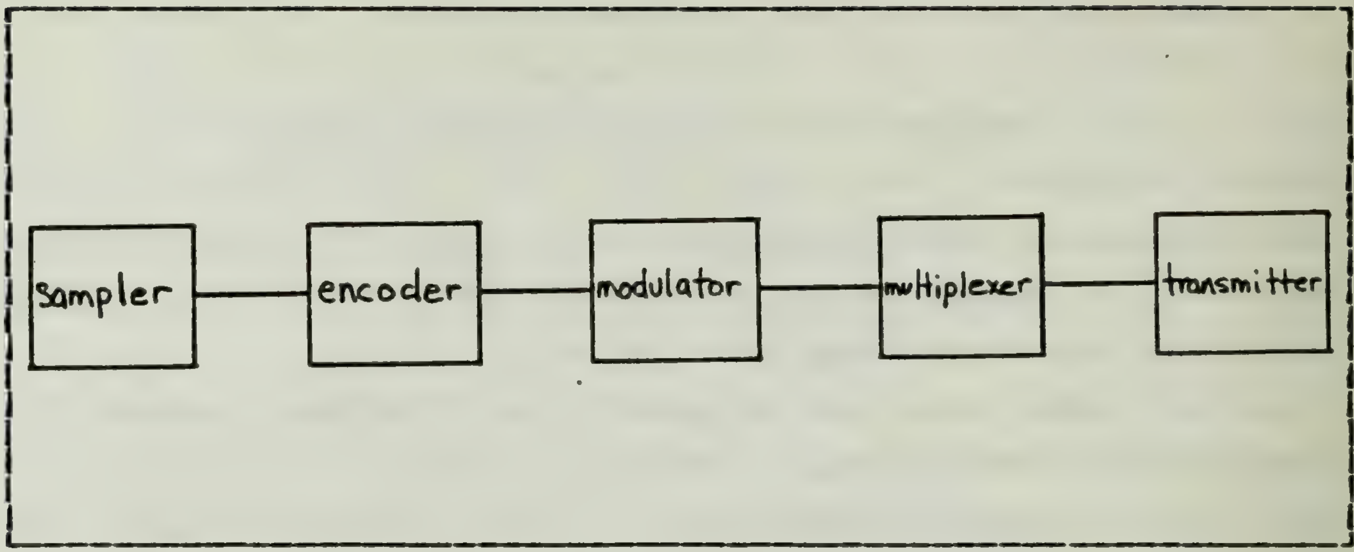



**Figure 5.1 Basic Hardware Component Block Diagram**

1. Sampler

The sampler is the device or component which samples or extracts those single characteristic values of a naturally occurring analog signal which are ultimately encoded into a digital word by the encoder. The frequency at which this sampler must operate was derived in the discussion of the Sampling Theorem. It was determined that the sampler must operate at a frequency greater than 2 times the highest significant Fourier frequency component of the source voltage. Modern solid state devices capable of taking thousands of samples/sec are available at modest costs.

2. Encoder

Often referred to as the A to D converter (analog to digital converter), the encoder transforms the samples of the analog signal derived from the sampler into a digital format through one of the analog to digital conversion techniques described in the chapter on A to D conversion. These digital bits can be stored for later use, coded, delayed or used immediately either individually or in groups to modify one of the characteristic qualities of the carrier waveform. When sample values of the analog source signal are converted into digital bits of information, they are simply that, digital bits of information. The carrier can only be modified to represent the source information by interaction with another voltage. Therefore the value of the digital bit (binary), either 0 or 1, is used to generate a baseband waveform or voltage which does the actual modulation of the carrier waveform.

a. Common Baseband Waveforms

It should be obvious that since it is desired to represent binary code words with a representative baseband waveform what is required is two levels of voltage. There are two basic logic schemes for representation of the baseband waveform. They are bipolar or unipolar logic. Bipolar logic involves representing the 0's or 1's of the binary codeword as either +V or -V as illustrated in Figure 5.2.

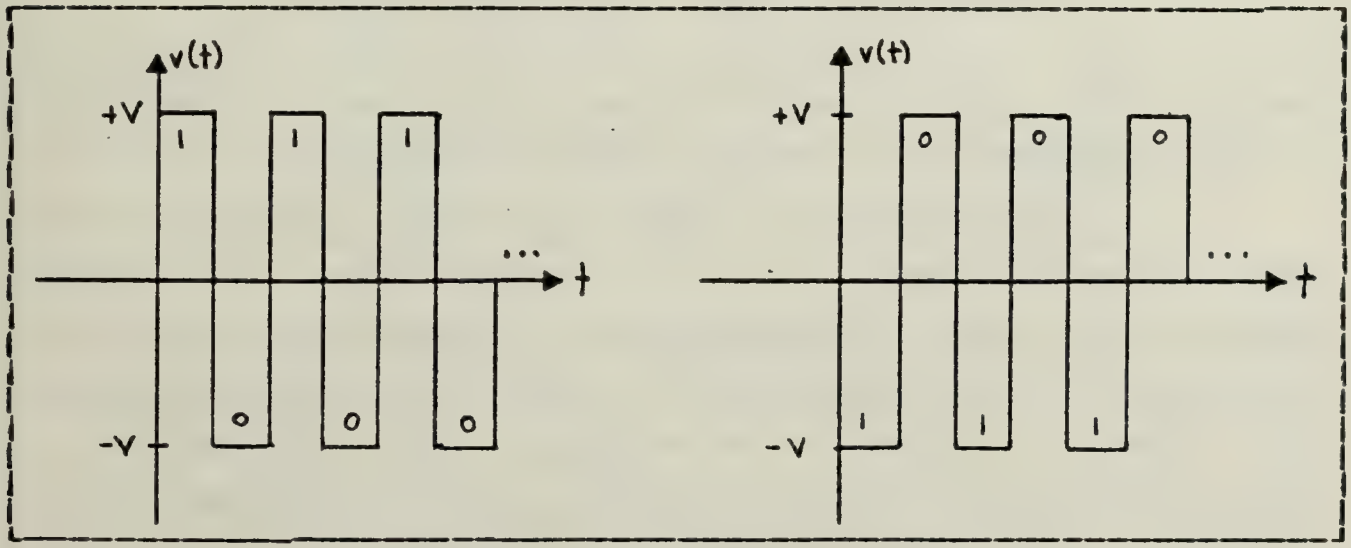



Figure 5.2 Bipolar Logic

The 0 or the 1 of the binary code word can be represented either as +V, -V or -V, +V respectively. This is a matter of convention but must be clearly understood in the various component designs in order to ensure compatibility between parts of the system.

Unipolar logic utilizes a voltage to represent either of the possible binary digits and the absence of voltage to represent the other. Two common conventions of unipolar logic are represented in Figures 5.3 and 5.4.

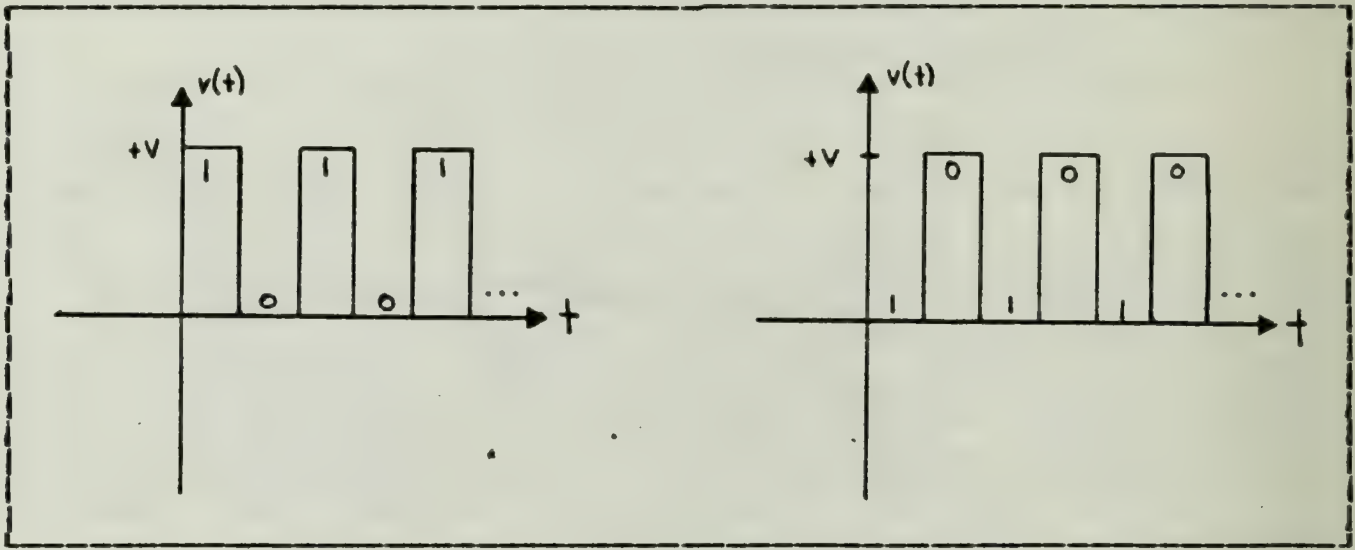



**Figure 5.3 Unipolar Positive Logic**

Again, which form of unipolar logic might be employed in the A to D converter is a matter of convention.

Variations of these two basic encoding formats have the advantages of ease of generation and improved decoding and clock recoverability. Two variations commonly used in satellite communications are the Non Return to Zero (NRZ) and the Manchester waveforms. The NRZ waveforms is simply the voltage representation which corresponds to the stream of bits represented in the logic scheme chosen. There is no transition as long as the same bit is present. Choosing bipolar logic as an example, a NRZ waveform can be described schematically as in Figure 5.5.

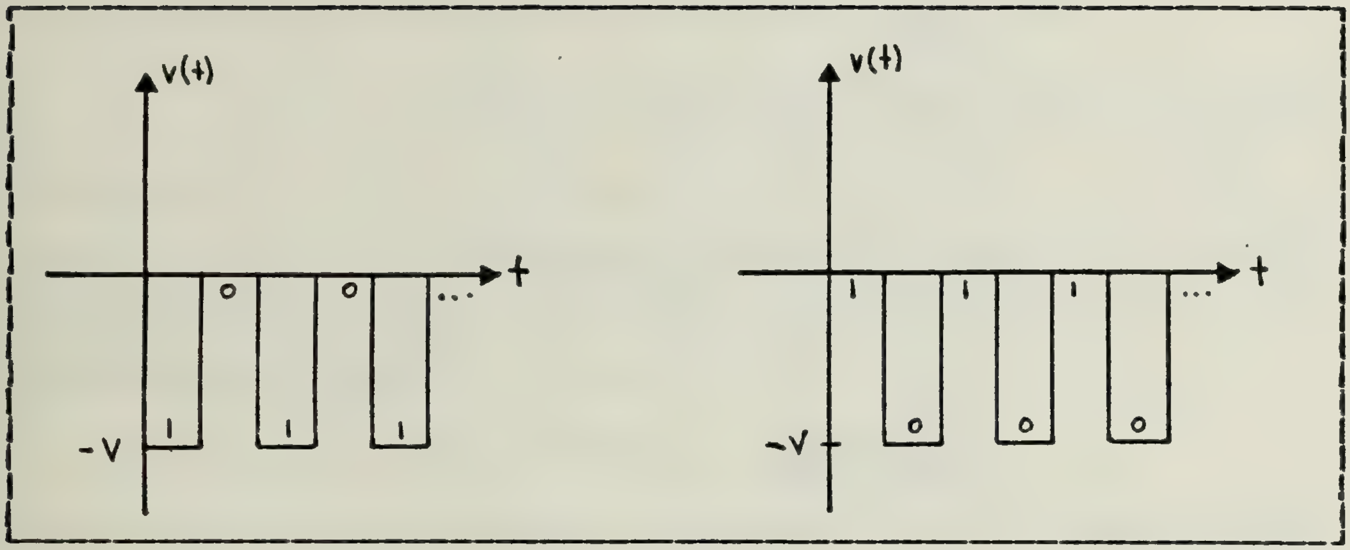



Figure 5.4 Unipolar Negative Logic

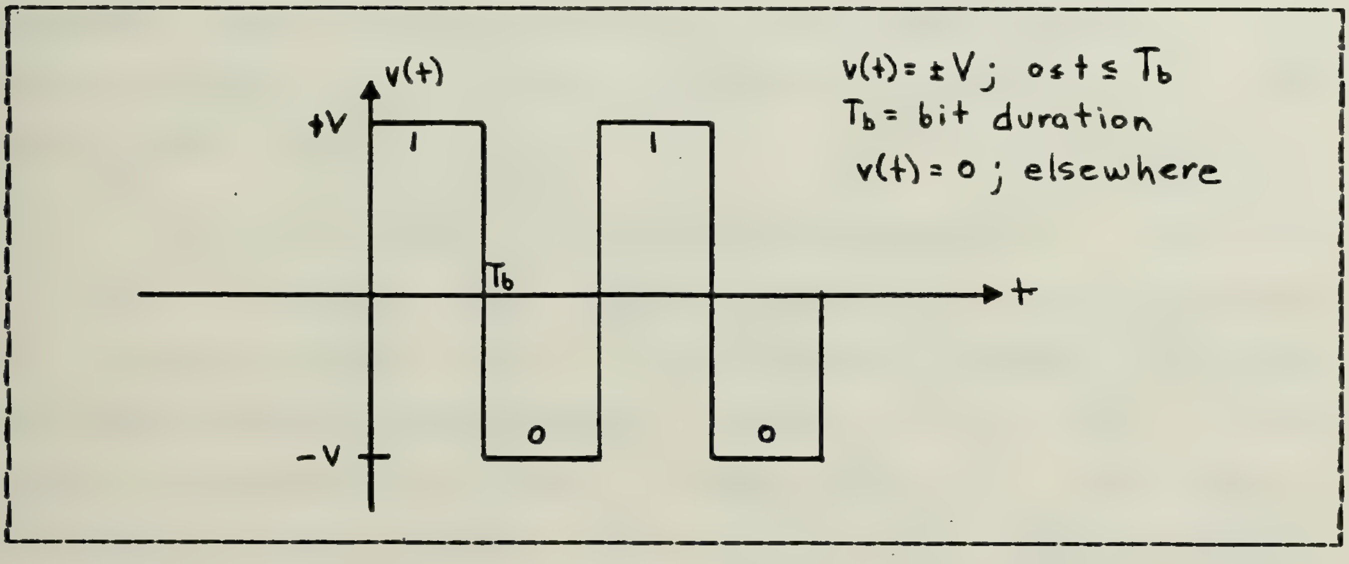



Figure 5.5 Non Return to Zero Waveform

The Manchester waveform of the same source voltage is represented with a transition at the midpoint of the bit duration from either +V to -V or -V to +V. This form of coding has the advantage of clock synchronization in all cases of digital encoding. Schematically, the Manchester code can be represented as in Figure 5.6.

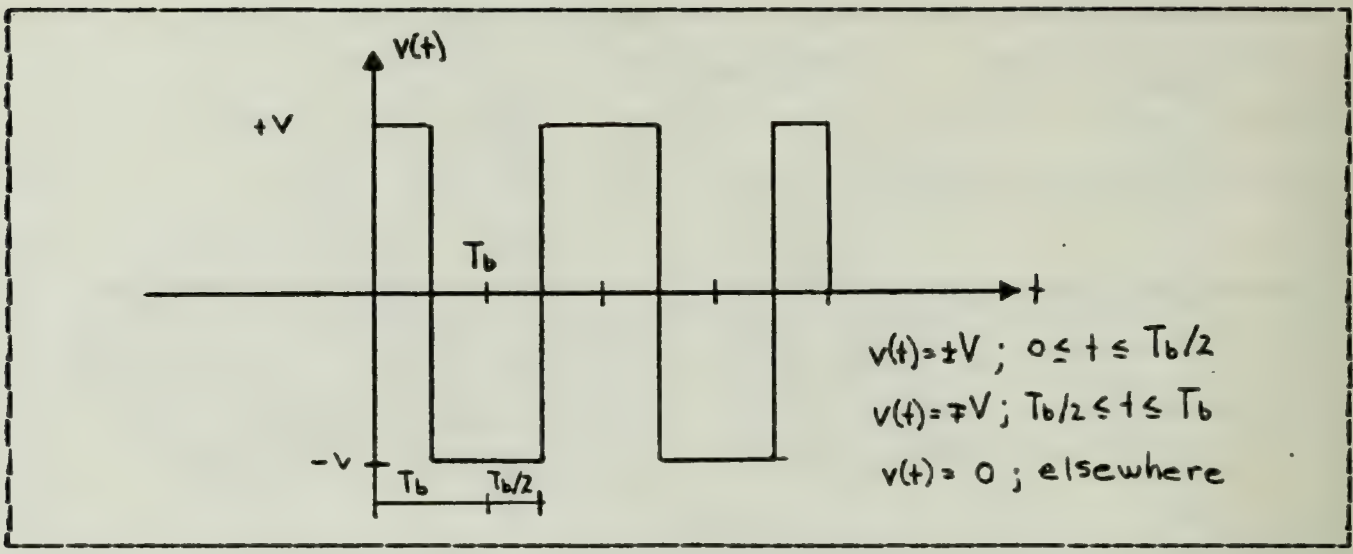



**Figure 5.6 Manchester Waveform**

### 3. Modulator

The modulator in a digital communications system is more commonly referred to as a modem (modulator/demodulator). This device does the actual transformation of the digital information into a waveform that can be transmitted from one point to another. The modulation as mentioned earlier can modify either the amplitude, phase or frequency of the carrier wave. In simple terms, the digital information derived from the analog source is mapped into the carrier wave. It is this mapping that determines the power and bandwidth characteristics of the modem.

### 4. Multiplexer

Multiplexing is a common method for increasing the utility of a given communications channel. Variations in multiplexing techniques give rise to the multiple access techniques employed in satellite communications.

a. Frequency-Division Multiplexing (FDM)

Frequency-Division Multiplexing is a technique whereby the capacity of a communications channel is increased by adding one signal to another. These signals occupy discrete nonoverlapping frequency bands. A specific signal is recovered by use of a band pass filter for the frequency band desired. [Ref. 7]

b. Time-Division Multiplexing (TDM)

Time-Division Multiplexing is a technique used to increase the capacity of a given communication channel by adding signals in time. That is, specific signals or portions of signals are allocated specific time slots or portions of the carrier wave. These time allocations cannot overlap and the signal is recovered through proper synchronization of the recovery hardware with the multiplexer. [Ref. 7]

c. Code-Division Multiplexing (CDM)

Code-Division Multiplexing is a technique for increasing the capacity of a given communications channel through the assignment of a characteristic code to the digital signal before it is multiplexed in the time or frequency domain. Since the code used in multiplexing is known at the demultiplexer, recovery of the digital source code could be accomplished through the use of a matched filter or correlator. [Ref. 7]

## C. BANDWIDTH

For purposes of discussion and for uniformity, when examining the various digital signal modulation techniques of interest, the bandwidth will be defined as the highest significant Fourier frequency component of the signal in question. For digital communications, it can be shown that by the above definition, the bandwidth of the baseband waveform is 1/Tb, where Tb is the bit duration. For example:

1. Voice sampled at the rate of 8000 samples/sec
2. Each sample represented by an 8 bit code word
3. Rb = bit rate = (8000 samples/sec)(8 bits/sample)
   Rb = 64 kbps
4. Tb = bit duration = 1/Rb
   Tb = 15.6 microsec

The Fourier transform of the baseband waveform of duration 15.6 microsec is represented in Figure 5.7.

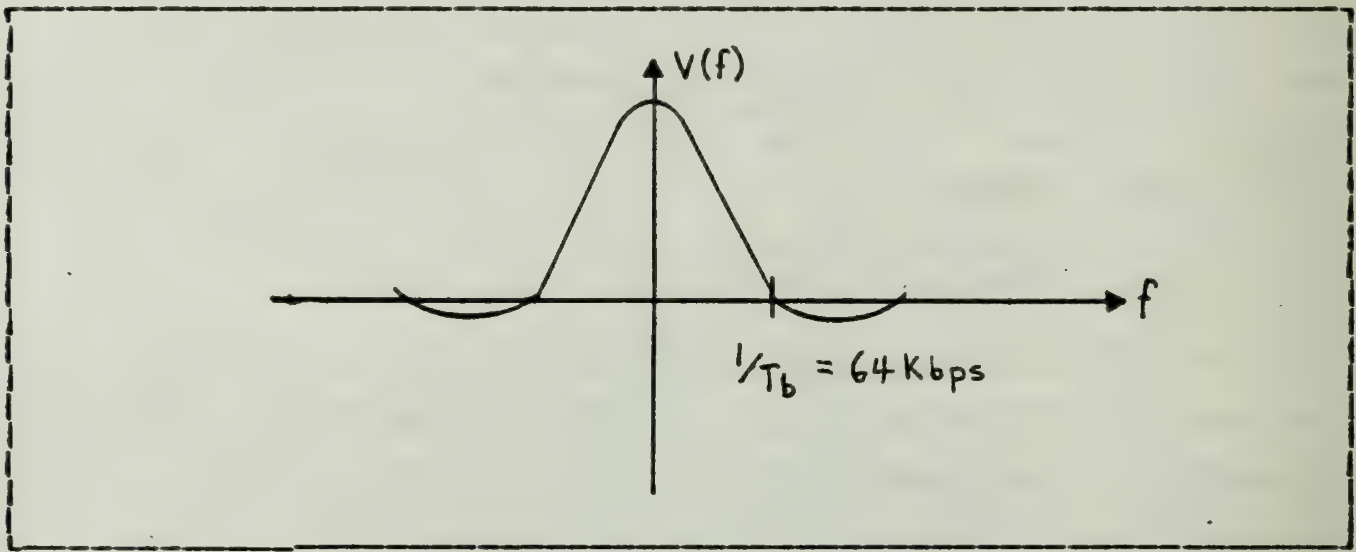



**Figure 5.7 Fourier Transform of Baseband Waveform**

It should be readily apparent that as the bit rate of the digital baseband waveform increases, so does the bandwidth. It is desireable to keep the bandwidth of the signal to a minimum due to limitations on availability of the electromagnetic spectrum. In addition, interference with signals which occupy adjacent frequency bands is reduced and propogation limitations of the medium and hardware limitations such as self-inductance and capacitance are minimized.

## D. SPECIFIC TECHNIQUES

Now with some of the basics of digital signals already introduced it is possible to investigate some of the more significant modulation formats employed in digital satellite communications. The organization of this section will be around the three general types of modulation, i.e., phase, amplitude and frequency. Within each broad category, several techniques will be described under the subareas of binary encoding, where the information is modulated onto the carrier bit by bit, and block encoding, where groups of bits do the modulation. Variations of these areas will be pointed out. The characteristics of the modulation technique will be introduced including, where practical, an analytical description, phasor representation, error probability, spectral efficiency, advantages and disadvantages.

### 1. Phase Shift Keying (PSK)

Phase shift keying is a generic form of signal modulation where the baseband waveform is used to modify or change the phase of the carrier. Phase, as stated earlier is one of the characteristics which distinguish one sinusoid from another. In general, phase shift keying offers the advantages of power and bandwidth efficiency.

#### a. Binary Phase Shift Keying (BPSK)

As the name suggests, Binary Phase Shift Keying (BPSK) results in a modulated carrier waveform consisting of two phases of the same sinusoid. The baseband waveform is used to change the phase of the carrier bit by bit from one phase to the other. In the case of bipolar logic, the bit stream consists of $\pm 1$. The general waveform of the carrier can be represented as a cosine wave of amplitude A, angular frequency w and initial phase angle delta as in equation 5.1.

$$vc(t) = A \cos(wt + delta) \qquad \text{(eqn 5.1)}$$

Since two phases are represented in vc(t), it is customary to let them differ by pi radians. This can be represented by the modification of equation 5.1 to account for a pi phase shift depending on whether the bit, v(t) to be modulated is either a +1 or -1 as shown in equation 5.2.

$$vc(t) = A \cos(wt + delta + pi/2(1-v(t))) \qquad \text{(eqn 5.2)}$$

Equation 5.2 can be expanded to result in a simplified form showing that the modulated carrier waveform is simply a product of the baseband waveform and the carrier. See equation 5.3 where v(t) represents the baseband waveform.

$$vc(t) = A\, v(t) \cos(wt + delta) \qquad \text{(eqn 5.3)}$$

A phasor diagram can be constructed to represent the modulated carrier waveform in BPSK and is shown in Figure 5.8. As can be seen by the phasor diagram in Figure 5.8, the phase of the modulated waveform depends on the value of the baseband waveform and differs by pi depending on whether the value of the modulated bit is ±1.

Remembering the section on the properties of the Fourier transform, it is possible to analyze the spectral and power efficiency of BPSK. Recall BPSK can be created by multiplication. Multiplication of two voltages is covered by the frequency translation property of the Fourier transform. Frequency translation results in double the bandwidth or spectral requirement of the translated waveform and half the power. In the case of a NRZ baseband waveform,

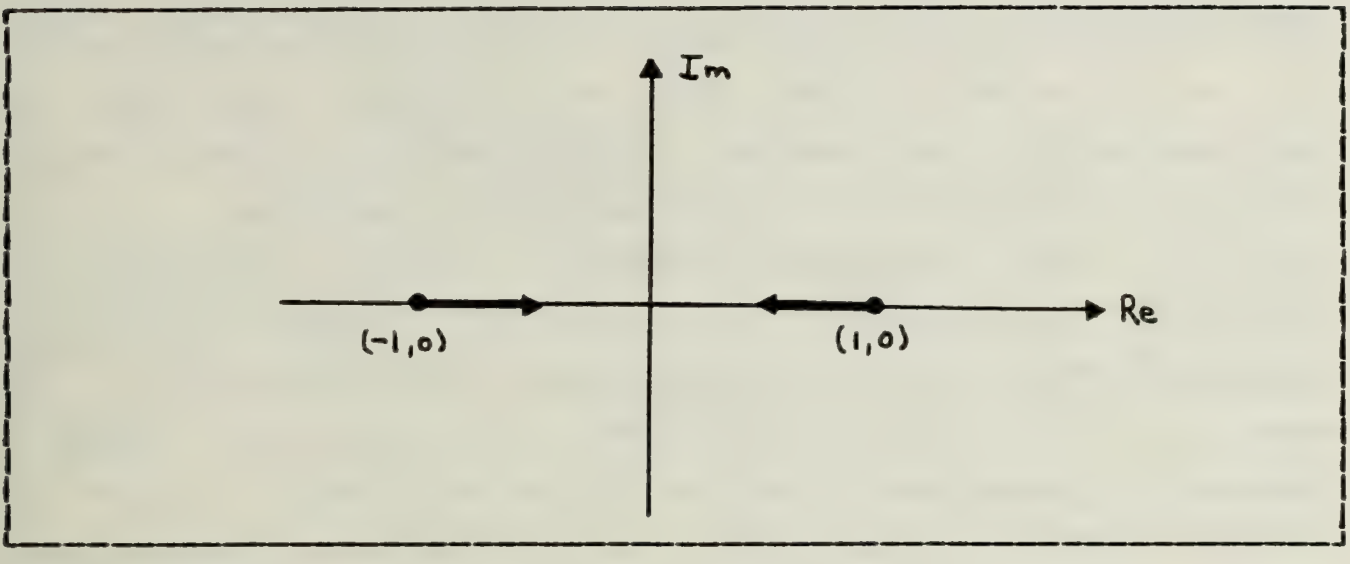



Figure 5.8 Phasor Diagram of BPSK

the bandwidth of the non-translated pulse, determined from the largest significant Fourier frequency component is 1/Tb, where Tb is the bit duration. The bandwidth of the translated voltage (BPSK carrier) is 2/Tb. The power in a sinusoid is $A^2/2$ and none of this power capacity is lost in BPSK since only the phase of the original sinusoid is changed. For a Manchester baseband waveform, the non-translated signal has bandwidth 2/Tb and the translated wave has bandwidth 4/Tb.

The advantage of BPSK is that it is relatively efficient in bandwidth utilization and power capacity. It does have the disadvantage of fairly large sidelobe components which contribute to interference with adjacent frequencies. This usually necessitates some type of filtering before the signal can be transmitted.

b. Differential Binary Phase Shift Keying (DBPSK)

The recovery of a Binary Phase Shift Keyed signal requires the use of a phase coherent reference signal as described in the previous sections on autocorrelation and matched filters. This may not always be practical or possible so a technique called Differential Binary Phase Shift Keying has be developed. DBPSK uses the same carrier type as BPSK and offers many of the same advantages and disadvantages. As stated, the primary difference is the lack of necessity for a phase coherent reference.

The baseband waveform, v(t), for DBPSK is generated by comparing the phase of the next bit to be transmitted to that of the previously transmitted bit. For example, if a +1 is encoded onto the carrier as v(t), it represents a particular phase of the carrier. If the next bit to be sent is also a +1, there is no phase change, i.e., the bit remains the same over two successive bit durations. If, however, the second bit to be sent is a -1, this represents a pi phase change and is decoded as a bit change over successive bit durations. This can be expressed mathematically as in equation 5.4 where bk represents the bit which will be encoded on the carrier and bk-1 represents the previous bit already encoded and ak represents the present bit.

$$bk = bk-1 \bullet ak \qquad \text{(eqn 5.4)}$$

As can be seen when bk = -1, there is a phase change and the bit changes from one bit to the next depending on the logic type being employed. If bk = +1, this represents the phase remaining constant over two successive bit durations, i.e., no bit change. See Figure 5.9 [Ref. 8]. Recovery is effected by correlation of the received bit to the previously received bit.

Differential Binary Phase Shift Keying (DBPSK) differs from Differentially Encoded Binary Phase Shift Keying (DEBPSK) in the recovery step only. DBPSK takes advantage of the modulation of phase shift in the recovery process while DEBPSK still utilizes a phase coherent carrier

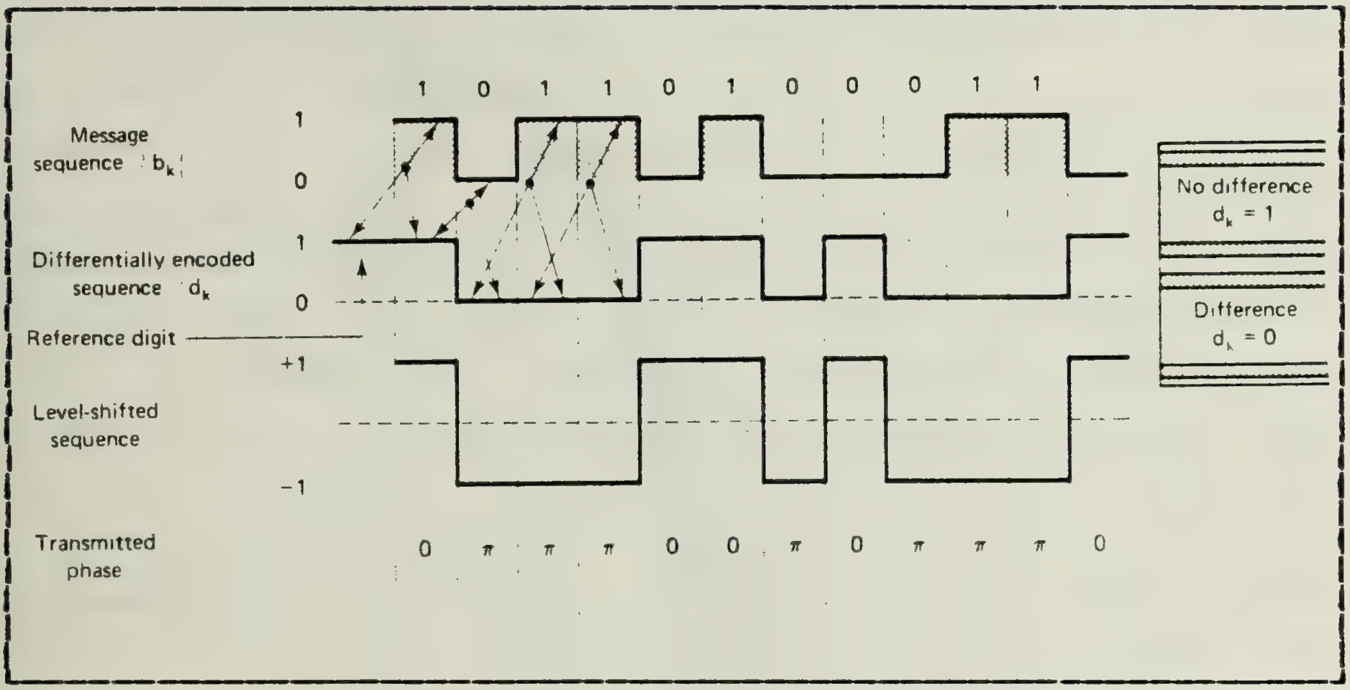



**Figure 5.9 Differential Binary Phase Shift Keying**

signal in demodulation. Both cases use the same scheme for differential encoding.

c. Orthogonal Binary Phase Shift Keying (Orthogonal BPSK)

In Binary Phase Shift Keying there is always a probability, with the interjection of noise on the transmitted signal that an individual bit will be decoded incorrectly from that which was actually transmitted. In some environments, this probability is so high that special encoding techniques must be employed in order to reduce the probability that a bit or a binary code word will be decoded in error. One such technique is Orthogonal BPSK. Orthogonal BPSK is a type of group encoding scheme where groups or blocks of bits from the original signal are assigned redundant bits according to a predetermined assignment routine. This new expanded sequence of bits is then transmitted as before in BPSK.

For example, assume that it has been determined that the respective values of the original signal will be represented by a k bit binary code word. It has been demonstrated that there exists 2**k binary code words, k bits long in this type of arrangement. With Orthogonal BPSK there also exists 2**k orthogonal sequences of bits (called channel symbols, bauds or chips) [Ref. 6], which represent the k binary code words. However, the number of bits or chips in the sequence is increased to n = 2**k, thereby providing the desired redundancy. Each of the 2**k sequences of n chips is constructed to be orthogonal to every other sequence in order to assure maximum separation. See Figure 5.10 [Ref. 6].

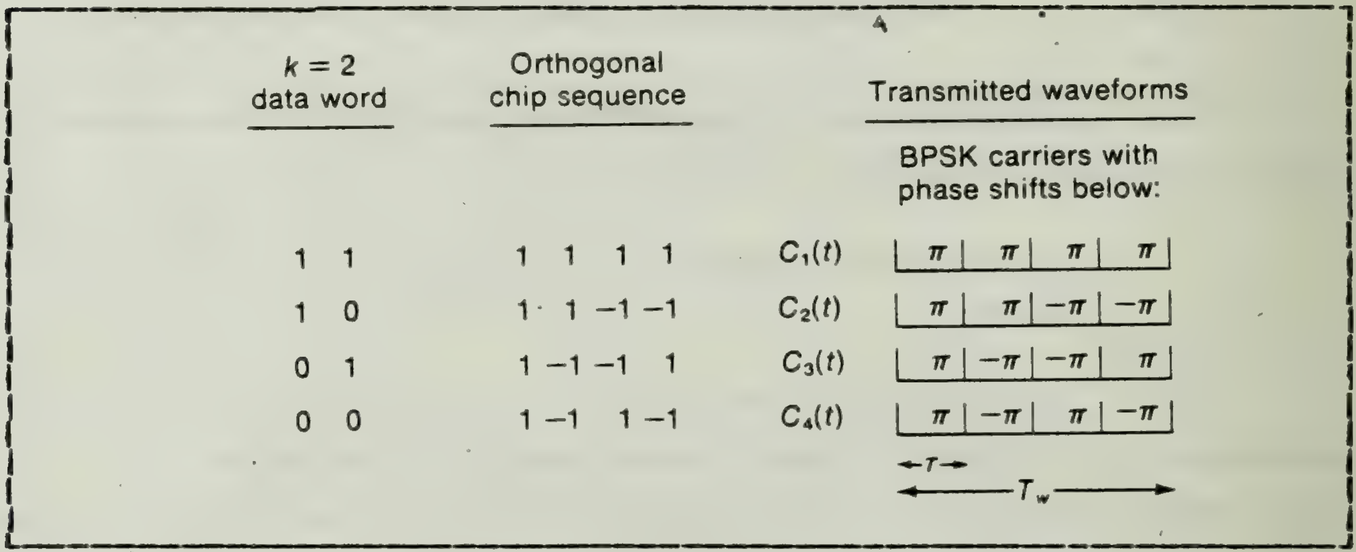



**Figure 5.10 Orthogonal Binary Phase Shift Keying**

Decoding is also done by blocks through a bank or 2**k correlators. Since the decoding is done in blocks, the probability that the entire transmitted code word will be incorrectly decoded is reduced even though individual bits are incorrectly decoded. This is the major advantage of Orthogonal BPSK. Figure 5.11 [Ref. 6], shows the reduction in error probability with Orthogonal BPSK

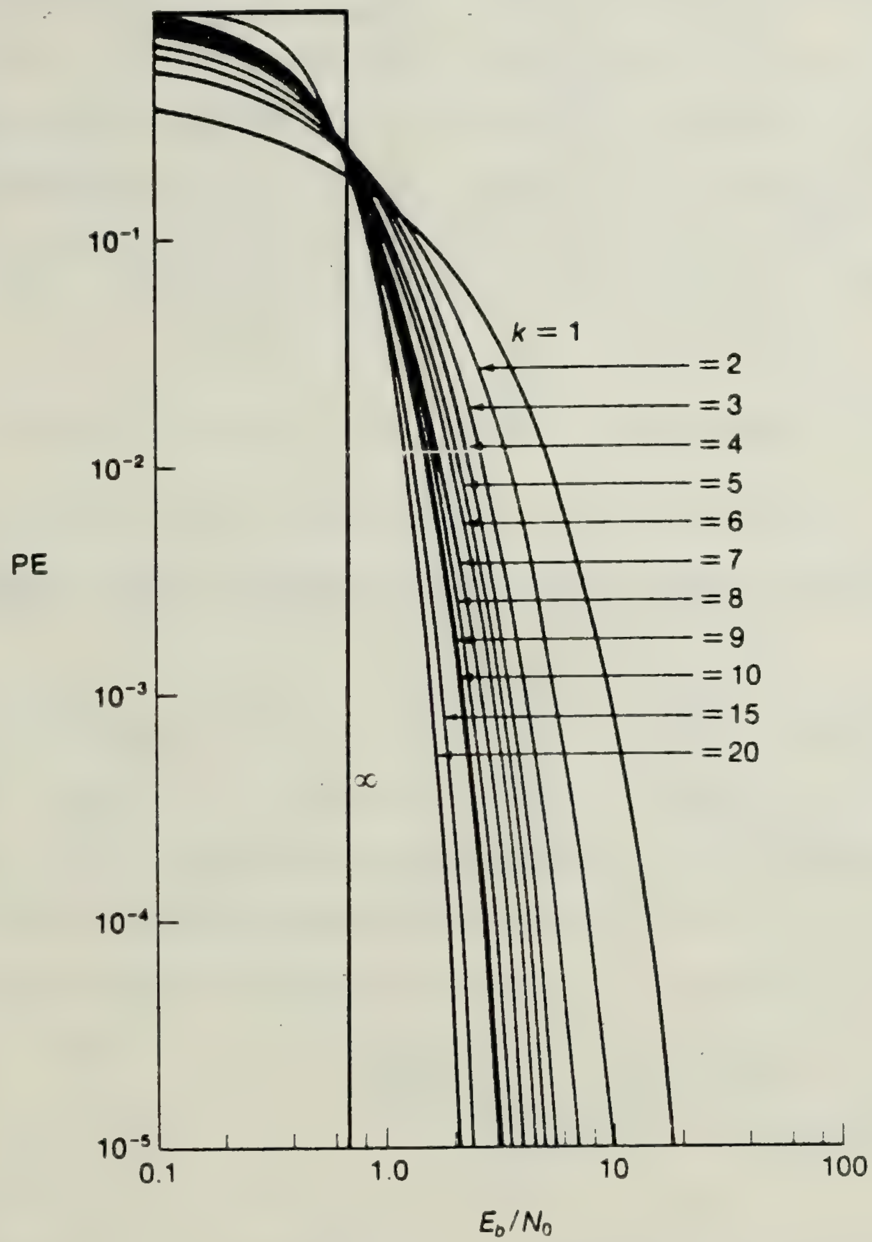



**Figure 5.11 Probability of Error for Orthogonal BPSK**

The major disadvantage to Orthogonal BPSK is that it results in a reduced data bit rate at a given transmission rate or the necessity for a higher transmission rate if the same bit rate is to be maintained. For example, suppose the bit rate of a PSK signal is R bits/sec. If each of the k bits in the PSK signal are represented by 2**k chips as is the case in Orthogonal BPSK modulation, then the bit rate must be (2**k/k)R if the same data rate is to be maintained. Additionally, since the necessary bit rate increases, the bit duration decreases. It was shown in the section on bandwidth that as bit duration decreases as the result of a higher bit rate, an increase in bandwidth is the result. Therefore, the tradeoff in decreased transmission error comes at the expense of bandwidth and bit rate. [Ref. 6]

d. Quadrature Phase Shift Keying (QPSK)

Quadrature Phase Shift Keying derives its name from a four-phase modulation of the carrier waveform. These modulations are achieved by simultaneously modulating the inputs from two bit streams onto a single carrier. The two bit sequences could be from two separate sources or successive bits from a single source. To take advantage of the orthogonality of the sine and cosine functions, QPSK can be viewed as in equation 5.5 as the sum of a BPSK modulated sine and cosine.

vc(t) = A v1(t) cos(wt + delta) + (eqn 5.5)
A v2(t) sin(wt + delta)

The signal can also be represented as in equation 5.6 where the phase angle, theta, of the modulated carrier is shown in equation 5.7.

vc(t) = 2**.5 A cos(wt + theta + delta) (eqn 5.6)

theta = arctan (v1(t)/v2(t)) (eqn 5.7)

The phases possible as a result of substitution of the possible bits in bipolar logic in equation 5.7 are ±45 degrees and ±135 degrees. See Figure 5.12.

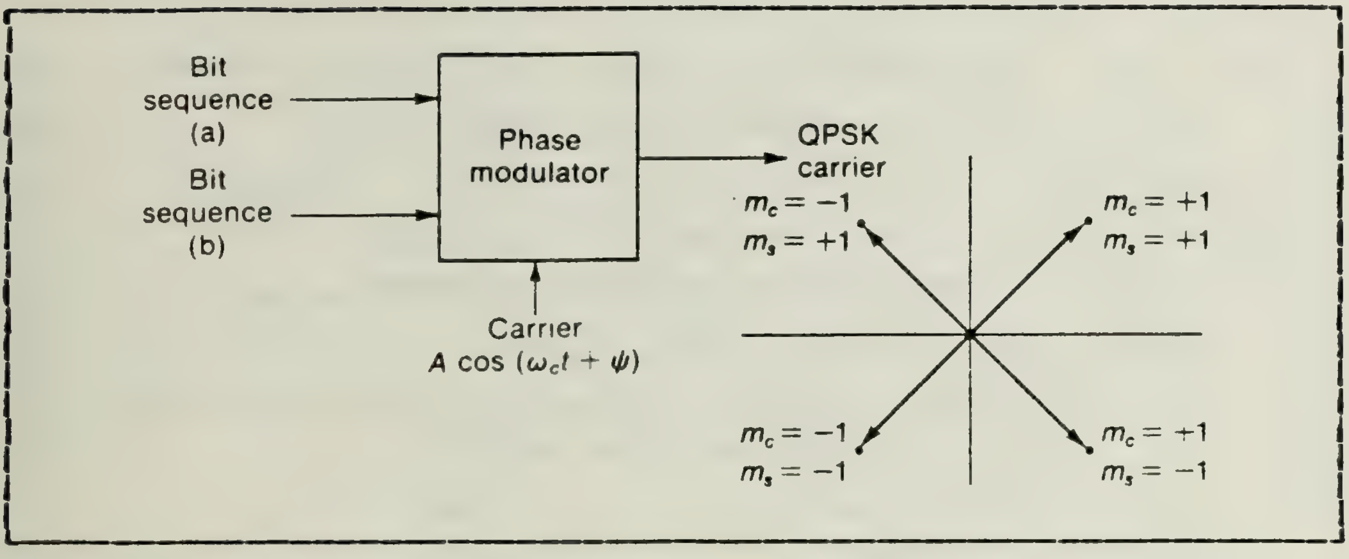



**Figure 5.12 Modulation and Phases of QPSK**

Since two bits are being encoded at one time for transmission, the bit rate in QPSK is twice that of BPSK. The BPSK bit rate is equal to the QPSK symbol rate, (symbol rate being the time the quadrature bits exist on the carrier). It is the symbol rate in QPSK which determines the bandwidth and since symbol rate is equal to bit rate in BPSK, the bandwidth of the two modulation techniques is identical. The great advantage of QPSK lies in the bandwidth utilization. Two bits of information are transmitted thereby effectively doubling the bit rate at no additional cost in bandwidth. A phasor representation of QPSK is illustrated in Figure 5.13. Each quadrature component of the modulated signal contains half the power of the total carrier or $(A^2/2)/2 = A^2/4$.

e. Offset Quadrature Phase Shift Keying (OQPSK)

In Quadrature Phase Shift Keying, both bits of the two baseband waveforms, which are to be modulated onto the carrier, change at the beginning of the respective symbol time. This allows the phase of the carrier to change up to 180 degrees for each symbol as illustrated in the

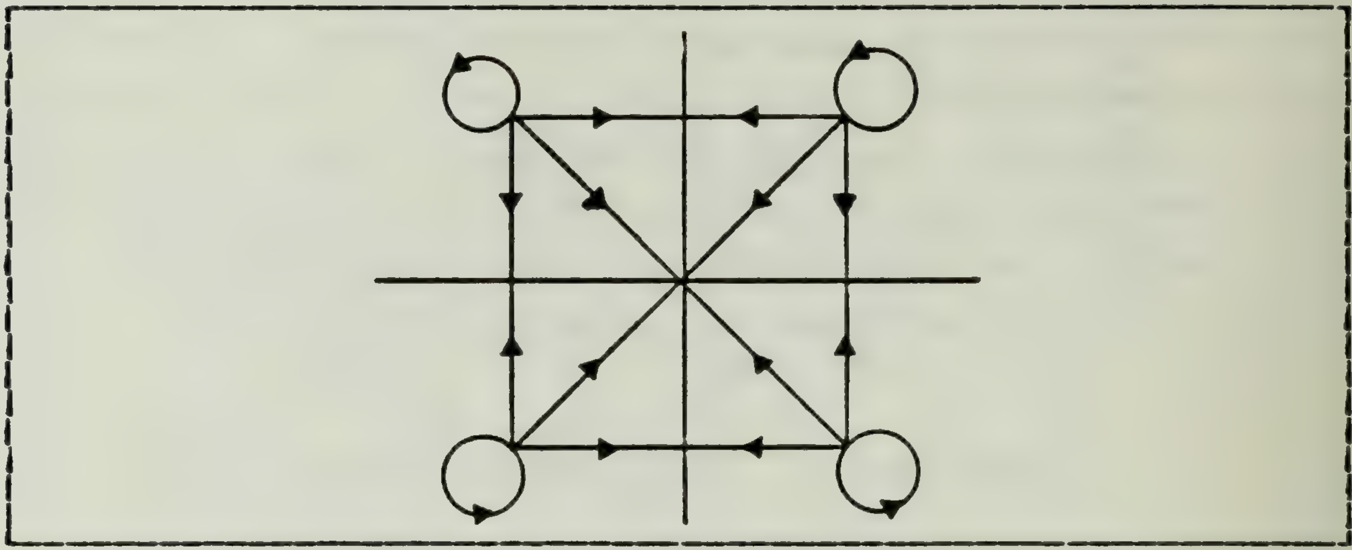


Figure 5.13 Phasor Diagram of QPSK

phasor diagram in Figure 5.13. The phase change of the modulated signal can be limited to 90 degrees through a modulation technique called Offset Quadrature Phase Shift Keying. This is accomplished by delaying the application of the second baseband channel for Ts/2 seconds, where Ts is symbol duration as illustrated in Figure 5.14. Modulation is then accomplished as before in QPSK.

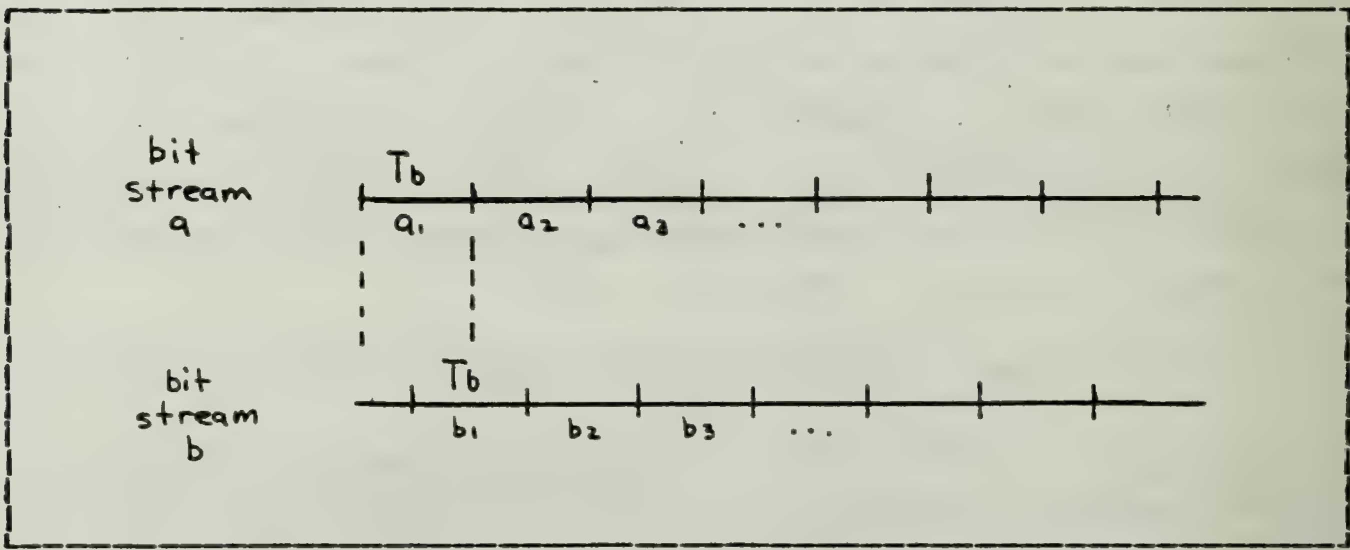



Figure 5.14 Offset Quadrature Phase Shift Keying

The result is a QPSK modulated signal in which the maximum phase shift between successive bits is 90 degrees. The bandwidth and power spectra of an OQPSK modulated signal are the same as that of a QPSK modulated signal. The advantage of OQPSK is in the limit which is placed on phase shift during encoding which simplifies the encoding hardware. Additionally OQPSK has spectral and interference advantages during decoding [Ref. 6].

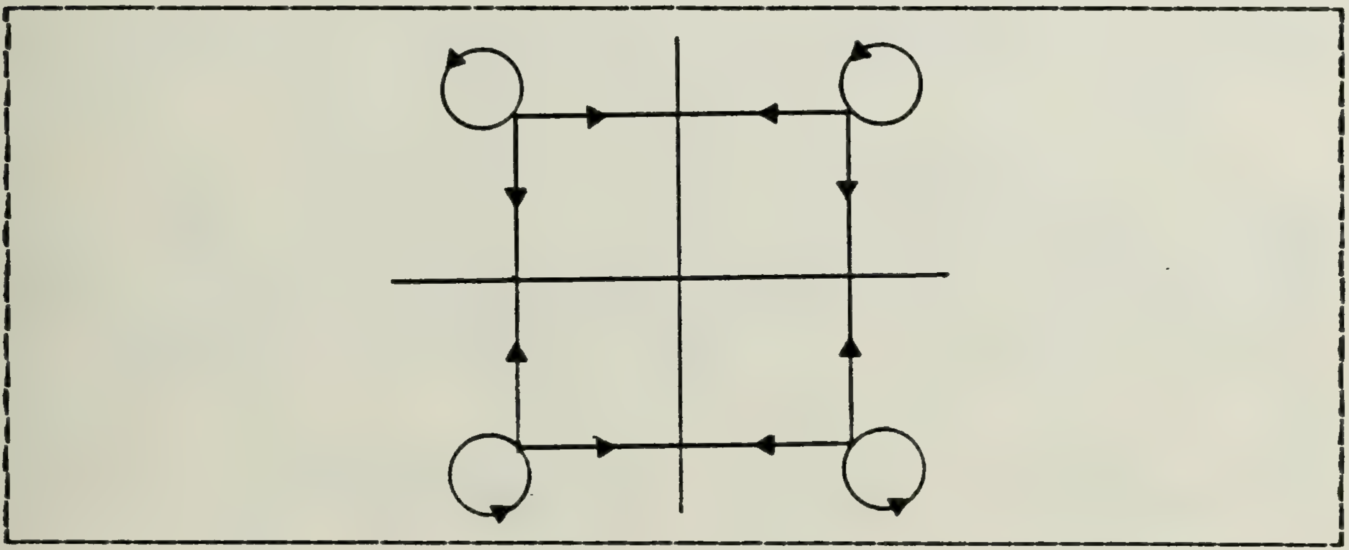


Figure 5.15 Phasor Diagram of OQPSK

f. Multiple Phase Shift Keying (MPSK)

Another form of digital modulation is known as Multiple or M-ary Phase Shift Keying (MPSK). Once again, as in Orthogonal Binary Phase Shift Keying, k data bits are transmitted. The k bits represent the word length and there are m = 2**k different binary words, k bits long, for this type of modulation. In MPSK, the binary code word is used to vary the phase of the carrier. There are therefore m = 2**k different phases in MPSK. These sequences of binary digits need not be orthogonal as in Orthogonal Binary Phase

Shift Keying. The length of the transmitted symbol is only k bits long as compared to 2**k bits for Orthogonal BPSK. The phases therefore are not all orthogonal leading to greater difficulty in decoding and a higher probability that an individual code word will be incorrectly decoded, especially as m becomes large.

As in other types of Phase Shift Keying, MPSK can be represented as in equation 5.8 where theta = phase = (2pi/m) i; i=0,1,...m-1.

$$vc(t) = A \cos(wt + theta + delta) \quad \text{(eqn 5.8)}$$

An advantage of MPSK is that as long as the symbol transmission rate does not increase, the carrier bandwidth does not increase even as the size of the binary code word increases. The disadvantage, as stated earlier, has to do with error rates in decoding. Decoding requires a phase coherent reference of considerable stability especially as m increases and respective phases become closer together. Practical limits for this type of modulation is m = 8 or a 3 bit binary code word due to the complexity of the encoding and decoding hardware.

2. Amplitude Shift Keying (ASK)

Amplitude Shift Keying, as the name implies is modulation of the carrier through variation of its amplitude due to variations in the baseband waveform. The information in the baseband waveform is thereby imparted to the carrier through this modulation. Since the amplitude of the carrier varies between successive symbols or binary code words, the power also varies. Recovery is by comparison of the transmitted power during each symbol to the possible m = 2**k power levels.

a. Multiple Amplitude Shift Keying (MASK)

MASK is a type of group encoding where k bits are combined into a single waveform for transmission and subsequent recovery. Since each symbol or binary code word contains k bits, there are m = 2**k different symbols and consequently m different amplitudes of the carrier possible during the symbol duration time. An assignment scheme which maps the binary code words into the m different amplitudes would be to simply space them Ai = A/mi volts apart, where A is the maximum amplitude and mi represents the decimal equivalent of the binary code word from i = 1 to 2**k.

The analytic form of a MASK waveform is represented in equaticn 5.9 where all the variables are the same a presented earlier except for Ai which is equal to one of i different amplitudes depending on the bit sequence to be transmitted.

$$vc(t) = Ai \cos(wt + delta) \qquad \text{(eqn 5.9)}$$

MASK is not very popular in satellite communications since it depends on a carrier of very stable amplitude. This is not generally practical under actual conditions. However, MASK does have the spectral advantage of a constant bandwidth of 2/Ts, Ts equal to symbol duration, even as the binary code word increases in length.

b. Quadrature Amplitude Shift Keying (QASK)

Quadrature Amplitude Shift Keying (QASK) is a hybrid digital signal modulaticn technique. It represents separate amplitude modulation of the quadrature components of a common carrier. In that sense it is both ASK and PSK. Implementation of QASK involves simultaneous application of M-ary Amplitude Shift Keying to the quadratures. See equation 5.10, where A1 and A2 are derived according to an assignment scheme as in MASK.

$$vc(t) = A1 \cos(wt + delta) + A2 \sin(wt + delta) \quad \text{(eqn 5.10)}$$

By this method, the effective bit rate is doubled over that of ordinary MASK. In other words, QASK results in the same data rate as modulating 2k bits onto the carrier at the same time or during the same symbol duration with MASK. The advantage to QASK lies in the fact that this increase in bit rate comes at no further increase in bandwidth. The disadvantages associated with MASK are still present with an additional increase in the complexity of the decoding equipment. Decoding of QASK must be done in two steps. First the signal is recovered in phase through a phase-coherent correlator and then each amplitude modulated signal is recovered as before in MASK by comparison of the power levels of the received signal.

### 3. Frequency Shift Keying (FSK)

Frequency Shift Keying (FSK), is the generic term used to describe a number of digital signal modulation techniques which cause variations in the carrier frequency by interaction with the baseband waveform. Decoding is generally through measurement of the frequency of the received signal.

#### a. Mimimum Shift Keying or Fast Frequency Shift Keying

Power requirement for the transmission or retransmission of a satellite signal is extremely important. It is desireable to limit power required to accurately transmit a piece of information to the minimum possible consistent with allowable error rates. For this reason, detection of FSK signals is limited to coherent methods as power required increases significantly for non-coherent methods.

MSK or FFSK are identical and represent a modulation technique known as continuous phase FSK. They get their names from the fact that "fast" indicates more bits per second can be transmitted in a given bandwidth than ordinary BPSK and "minimum" refers to the minimum modulation index for which orthogonal signalling occurs [Ref. 7].

Analytically, FFSK can be expressed as in equation 5.11 and equation 5.12.

$$vc(t) = A \cos((wc + deltaw)t) \qquad \text{(eqn 5.11)}$$

$$vc(t) = A \cos(\pm deltaw\ t) \cos(wc\ t) - A \sin(\pm deltaw\ t) \sin(wc\ t) \qquad \text{(eqn 5.12)}$$

As is noted, FFSK can be envisioned as the summation of separately modulated in-phase and quadrature components of the carrier by the baseband waveform. The technique is similar to that used in OQPSK. The frequency variation in the carrier waveform is between wc + deltaw and wc - deltaw. Maximum separation in phase of these two frequency components occurs at pi as noted in equation 5.13, where Tb represents the bit or symbol duration.

$$(w2-w1)Tb = pi \qquad \text{(eqn 5.13)}$$

This can be converted to the modulation index mentioned above simply by converting to frequency as in equation 5.14.

h = (f2-f1)Tb = .5 (eqn 5.14)

For FFSK, the frequency deviation or separation between frequencies of the transmitted carrier is exactly 1/2Tb and the deviation from the carrier frequency is 1/4Tb. FFSK can therefore be rewritten from equation 5.12 as in equation 5.15 by substitution for deltaw = 2pi(1/4Tb).

vc(t) = A cos(±pi t/2Tb) cos(2pi fc t) - (eqn 5.15)
A sin(±pi t/2Tb) sin(2pi fc t)

The error rate in this modulation technique is identical to that of BPSK. Frequency Shift Keying can be implemented to incorporate any given modulation index by simple calculation of the frequency separation with equation 5.14 and substitution into equation 5.15. For M-ary Frequency Shift Keying, the separation between frequencies must be at least 1/Ts, where Ts is the symbol duration, in order to avoid carrier energy from one frequency being incorrectly interpreted as carrier energy from another frequency. This results in a modulation index of 1. Figure 5.16 represents the phasor diagram of an FFSK modulated signal.

4. <u>Quadrature Partial Response Signalling</u> (<u>QPRS</u>)

Quadrature Partial Response Signalling (QPRS) represents a specific type of the general class of digital signal modulation techniques called Partial Response Signalling. Partial Response Signalling is a method in which a controlled amount of intersymbol interference (ISI) is allowed during the encoding process. Previously any overlap in successive signals resulted in aliasing and an increased probability of error in decoding. The idea was

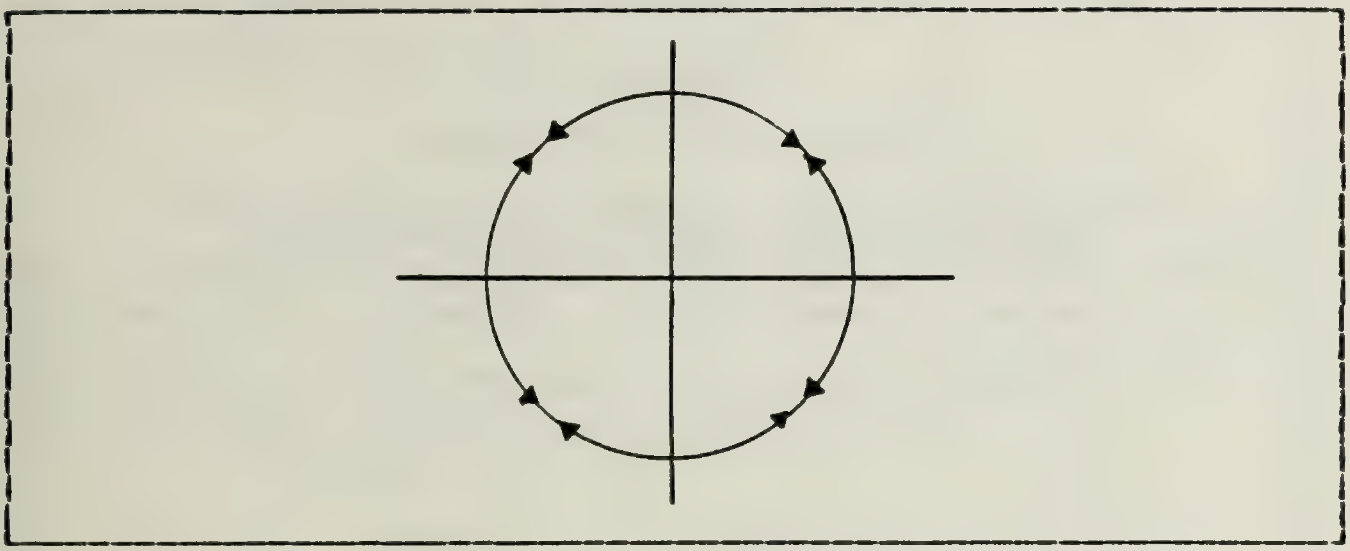


**Figure 5.16 Phasor Diagram of FFSK or MSK**

first introduced by Lender in 1963 [Ref. 12]. His concept was that in knowing and controlling the amount of interference allowed during the encoding process, compensations can then be made for the ISI at the receiver. Allowing for a limited amount of ISI makes it possible to transmit at rates equal to the Nyquist rate, something that is not possible with many other systems [Ref. 13].

QPRS is implemented, as with previous quadrature systems, through simultaneous modulation of the quadrature components of a common carrier. However, this time the modulation is accomplished with a PRS system. The modulation of the respective quadrature components represents the impulse response of one of a number of linear filters to the bits of the baseband waveform. The linear filter employed depends on the class of the Partial Response Signalling system being used. There are 5 classes of linear filters commonly used in PRS and they are illustrated analytically in Table 1 and graphically in Figure 5.17.

Output of a QPRS system can be represented as in equation 5.16, where an and bn represent the nth bit to be modulated and h(t - nTs) represents the contribution of the

## TABLE 1

## QPRS SYSTEM POLYNOMIALS

| SYSTEM POLYNOMIAL $F(D)$ | FREQUENCY RESPONSE $H(\omega)$ for $\lvert\omega\rvert \le \pi/T$ | IMPULSE RESPONSE $h(t)$ |
|---|---|---|
| $1 + D$ | $2T \cos \frac{\omega}{2} T$ | $\frac{4T}{\pi} \frac{\cos(\pi t/T)}{T^2 - 4t^2}$ |
| $1 + 2D + D^2$ | $4T \cos^2 \frac{\omega}{2} T$ | $\frac{2T^3}{\pi t} \frac{\sin(\pi t/T)}{T^2 - t^2}$ |
| $2 + D - D^2$ | $T + T \cos \omega T + j3T \sin \omega T$ | $\frac{T^2}{\pi t} \sin(\pi t/T)(\frac{3t-T}{t^2-T^2})$ |
| $1 - D^2$ | $j2T \sin \omega T$ | $\frac{2T^2}{\pi} \frac{\sin(\pi t/T)}{t^2 - T^2}$ |
| $1 - 2D^2 + D^4$ | $-4T \sin^2 \omega T$ | $\frac{8T^3}{\pi t} \frac{\sin(\pi t/T)}{t^2 - 4T^2}$ |

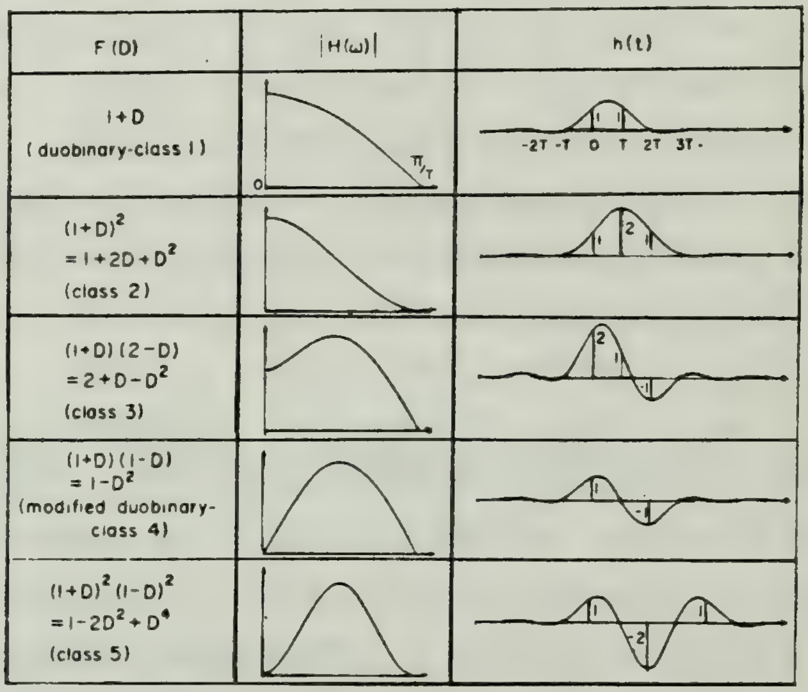



Figure 5.17 QPRS Linear Filter Impulse Responses

impulse response of the linear filter at time t for bit n. [Ref. 14]

$$v_c(t) = \sum_{n=-\infty}^{\infty} a_n \, h(t - n T_s) \cos(wt + delta) \qquad \text{(eqn 5.16)}$$
$$\sum_{n=-\infty}^{\infty} b_n \, h(t - n T_s) \sin(wt + delta)$$

QPRS has the advantage of rapid transmission rates with no increase in bandwidth and excellent error handling performance. The cost of such performance improvement comes in the form of a higher required signal to noise (SNR) ratio when compared to other binary systems.

# VI. COMPUTER SIMULATION OF DIGITAL SIGNAL MODULATION TECHNIQUES

Due to the nature of digital signal modulation, it is possible to simulate systems through the use of the digital computer. This chapter is a description of the computer simulation included in the Appendix of this thesis.

## A. GENERAL DESCRIPTION OF THE PROGRAM

The program was constructed using top-down design and modular programming. FORTRAN was selected as the programming language due to its capabilities in the area of numerical operations and the availability of mathematical routines to be used as modules in the program. Testing was accomplished on each module as the program was developed. The program was designed to be used interactively in order to allow for instructional use. Although the program was made user friendly in that error checking is accomplished on user inputs, care must be taken to insure instructions are followed correctly.

The program consists of twenty modules, sixteen of which were written by the author, three which were taken from the Double Precision International Mathematics and Statistics Library (IMSLDP) and one which was taken from a NON-IMSL library which resides on the IBM 3033 located at the Naval Postgraduate School. The IBM 3033 was used exclusively for development and testing.

The development was accomplished using the WATFIV compiler; however, the program was written to operate with the VS FORTRAN compiler as well. Testing and operation was done using the VS FORTRAN compiler. Figure 6.1 is a block diagram of the relationship of the modules for a representative modulation subroutine.

## B. MAIN CONTROL MODULE

MAIN operates as the control module for the entire computer simulation and accomplishes limited output. It introduces the program and allows the user to input various parameters of the digital signal modulation technique to be simulated and various other control functions. These inputs include:

1. Modulation technique
2. Class of QPRS system (when appropriate)
3. Digital logic scheme
4. Baud or symbol rate
5. Bits/binary code word (when appropriate)
6. Carrier frequency
7. Number of samples to be generated
8. Carrier max amplitude
9. Initial phase angle
10. Use of random number generator or no
11. Seed for RNG (when appropriate)
12. Number of repetitions of the simulation

Once the user has input the desired characteristics of the modulation to be simulated, MAIN calls the appropriate module to begin the actual calculation. The values necessary to calculate the time series of the signal are passed to the subroutines as parameters. MAIN calls the required subroutine the number of times the user specifies as repetitions. Each call to a subroutine produces the Discrete Fourier Transform and amplitude spectrum of the signal. It is these respective amplitude spectrums which produce the statistics in the final output after MAIN calls the statistics subroutine the final time.

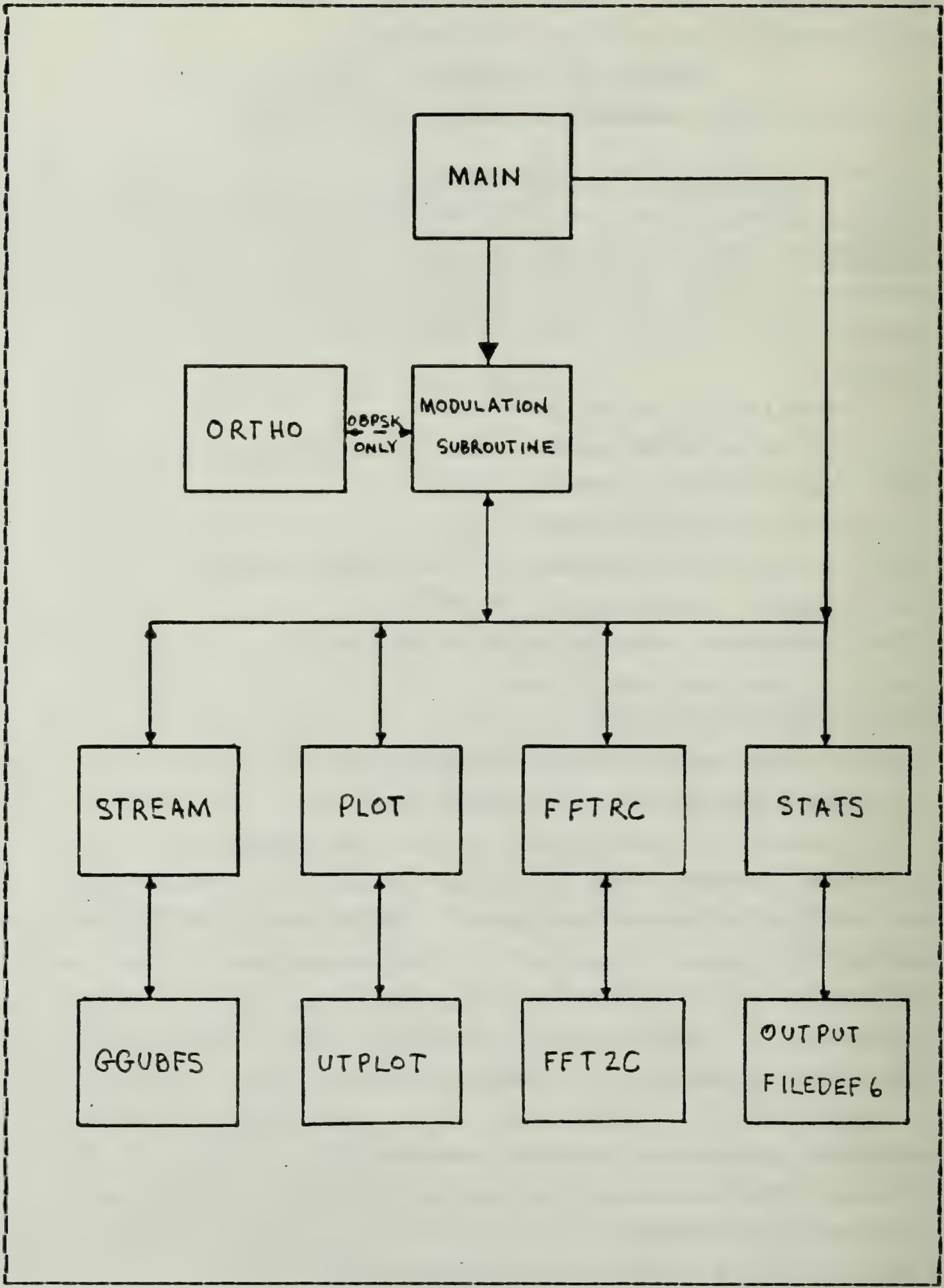



Figure 6.1 Representative Block Diagram

## C. SUBROUTINE BPSK

This description of SUBROUTINE BPSK, a module that simulates Binary Phase Shift Keying, will serve as a guide on the construction and operation of subsequent modules involved in the actual signal modulation. A complete description including its interaction with other program subroutines will be included. Since other signal modulation subroutines interact in the same manner within the program, subsequent descriptions will concentrate on differences to this basic module.

SUBROUTINE BPSK begins with variable declarations as with all FORTRAN programs. All variables are declared to be either type integer or double precision. All calculations in the program are carried out in double precision accuracy.

Variable initialization is next. Significant variable initialized at this point are TIME, the time of the first sample which is set to 0. STEP, the delta time between samples is set to the normalized Nyquist sampling rate. OMEGA, the angular frequency and DELTA, the initial phase angle are computed in radians. NBAUD, a variable which keeps track of the number of symbols which have been modulated, is set to 1 or the first symbol is being modulated. NBAUD remains at its present value until total elapsed time exceed 1 symbol duration or BAUDD, the normalized baud rate.

Subsequently, an initial call is made to SUBROUTINE STREAM to receive the value of the first bit to be modulated. The modulation is then carried out according to equation 6.1.

The values of each sample are assigned to an array for storage along with the corresponding time increment. Modulation continues with the first drawn bit until 1 bit or symbol duration is exceeded when STREAM is called to draw vc(t) = A v(t) cos(wt + delta) (eqn 6.1)

A = amplitude
v(t) = bit to be modulated
w = angular frequency
t = time
delta = initial phase angle

another bit. This entire process is repeated until the number of samples specified by the user in MAIN is reached.

At this point, only on the first repetition of the simulation, SUBROUTINE PLOT is called which gives the user the opportunity to plot the time series of the simulated signal. Upon return from PLOT the subroutine calls the IMSL routine FFTRC to generate the Discrete Fourier Transform of the time series produced. This is accomplished on each repetition of the program.

Again on the first repetition only, information on the number of the principle harmonic of the FFT is displayed. Plot is called again to give the user the opportunity to plot the amplitude spectrum. The values of the FFT and the amplitude spectrum are computed on each successive repetition of the program and the amplitude spectrum is added to the statistics being accumulated by a call to SUBROUTINE STATS.

## D. SUBROUTINE DBPSK

SUBROUTINE DBPSK simulated Differential Binary Phase Shift Keying and is essentially the same as SUBROUTINE BPSK. The difference lies in the fact that what is encoded during the modulation process is the product of the random bit drawn and the value of the bit previously modulated. A reference bit equal to +1 is used to determine the first bit to be modulated. In this manner, modulation of +1 means no change occurs between successive bits and the phase of the carrier remains the same. Modulation of a -1 indicates a change of bits has occurred and is indicated by a change of phase of the carrier between successive bit durations. The rest of the module remains unchanged.

## E. SUBROUTINE OBPSK

Orthogonal Binary Phase Shift Keying is accomplished in this module. Again SUBROUTINE OBPSK is constructed essentially the same as SUBROUTINE BPSK. In Orthogonal BPSK the only bit streams that are modulated are n bits long representing each binary code word, where n = 2**k (k = number of bits per word is limited to 6 bits in MAIN). Each sequence of n bits is orthogonal to every other allowed sequence.

The way this is accomplished in this module is first to draw a series of k random bits from SUBROUTINE STREAM. This series of k 1's and 0's is then changed to its decimal equivalent from 0 to 2**k-1 and saved for later use.

The program continues with a call to SUBROUTINE ORTHO which generates an n X n orthogonal matrix of +1's and -1's through the use of the Hadamard matrix and the Kronecker product [Ref. 15]. The Hadamard matrix is a 2 X 2 orthogonal matrix of 1's and -1's. The Kronecker product is the matrix which is formed when a matrix is expanded to twice its original size by replacing each element of the Hadamard matrix by the product of the Hadamard matrix and the original matrix. The value of the decimal equivalent to the k random bits is used to identify the row of the matrix and the respective columns represent the value of the binary digit which is modulated onto the carrier by BPSK. This sequence continues until either one bit duration is exceeded, when another orthogonal bit is received from ORTHO or when all the orthogonal bits in a row are modulated. Then another sequence of k random bits is converted to its decimal equivalent and a new row and column of the orthogonal matrix is identified. Execution continues in this manner until all required samples have been produced. It should be noted that the bit rate for Orthogonal BPSK is n times the baud rate. This significantly increases the signal bandwidth requirement and decreases the time increment between successive samples in the simulation. The rest of the module is the same as those previously presented.

## F. SUBROUTINE QPSK

Quadrature Phase Shift Keying, simulated in SUBROUTINE QPSK, is accomplished through simultaneous BPSK modulation of the quadrature components of the carrier. Two random bits are drawn from successive calls of SUBROUTINE STREAM and each bit is in turn modulated onto the carrier components and the sum formed. This process is repeated at time intervals equal to the normalized Nyquist sampling rate until TIME exceeds one symbol duration, BAUDD. At this time two new random bits are drawn and the modulation continues until all required samples are produced. The rest of the modules is the same as those before. Equation 6.2 represents the analytical expression used to simulate

## G. SUBROUTINE OQPSK

SUBROUTINE OQPSK is essentially the same as SUBROUTINE QPSK with these minor deviations. The time of the first sample is artificially initialized as TIME = .5(BAUDD) or 1/2 the symbol duration. In this manner, the two random

vc(t) = A v1(t) cos(wt + delta) + (eqn 6.2)
A v2(t) sin (wt + delta)

v1(t) = in-phase bit
v2(t) = quadrature bit

bits first being modulated are both known to have existed on the carrier for 1/2 the symbol duration. The first of thest two bits only remains on the carrier until TIME = 1(BAUDD) when a third random bit is drawn and modulation begins with this bit. The second random bit is allowed to remain until TIME = 1.5(BAUDD) when a fourth random bit is drawn to replace it and so on. In this manner the two bits being modulated are offset in the time they change by 1/2 the symbol duration. This necessitates keeping track of the number of bits modulated on both the in-phase and quadrature component of the carrier. The remaining portion of the module is the same as SUBROUTINE QPSK.

## H. SUBROUTINE MPSK

This module simulates M-ary Phase Shift Keying. This is accomplished by changing the phase of the carrier to any one of n =2**k phases where k is the number of bits in the binary code word. The max length of the binary code word is limited to 10 in the MAIN program. This modulation is accomplished by use of equation 6.3 to simulate the MPSK system.

The program first draws k random bits and converts them to the decimal equivalent. At this point modulation begins at TIME = 0 according to equation 6.3. Modulation continues until one symbol duration is exceeded when a new set of k random bits is drawn, conversion to decimal takes place and

$$vc(t) = A \cos[wt + (2pi\ m)/n + delta] \qquad \text{(eqn 6.3)}$$

m = decimal equivalent of binary code word

n = 2**k; k = bits in binary code word

modulation continues or until all desired samples have been produced. The program continues as in previous subroutines.

## I. SUBROUTINE MASK

The principles of operation of SUBROUTINE MASK is similar to SUBROUTINE MPSK which also involves a group encoding system. SUBROUTINE MASK allows for M-ary Amplitude Shift Keying. The analytic expression used in the simulation is provided in equation 6.4 Ai is one of 2**k equally spaced amplitudes.

$$vc(t) = Ai \cos(wt + delta) \qquad \text{(eqn 6.4)}$$

Once again a set of k random bits is drawn from SUBROUTINE STREAM. The decimal equivalent of the k bits is computed and used to adjust the amplitude Ai according to the assignment routine expressed in equation 6.5.

$$Ai = A/m \qquad \text{(eqn 6.5)}$$

A = max amplitude

m = 1 to n; n = 2**k

This process is repeated at each symbol duration until all desired samples have been produced. The rest of the module remains unchanged.

## J. SUBROUTINE QASK

SUBROUTINE QASK combines the elements of the previously described amplitude modulation technique with the now familiar quadrature modulation technique. Two separate streams of random numbers are generated and converted to their decimal form. These numbers are used to produce two separate amplitudes with the same assignment scheme used in MASK. These amplitudes modulate the quadrature components of a common carrier and the output is formed by the sum of the quadratures. Modulation continues in the above manner until a symbol duration elapses when two new amplitudes are calculated. Processing terminates when all required samples are computed.

## K. SUBROUTINE MSK

Minimum Shift Keying is frequency shift keying in which the modulation index is 1/2. The modulation index is represented in equation 6.6

h = .5 = (deltaf) Tb (eqn 6.6)

deltaf = frequency separation

Tb = bit duration

The frequency deviation from the carrier is .5(deltaf) = 1/4Tb. Converting this frequency deviation to radians yields pi/2Tb. This angular frequency deviation from the central carrier is either ±pi/1Tb depending on the value of the bit to be modulated.

The simulation in SUBROUTINE MSK is accomplished by drawing a random bit and generating the signal according to equation 6.7.

$vc(t) = A \cos[(w \pm pi/2Tb)t + delta]$ (eqn 6.7)

Once again modulation continues until one bit duration is elapsed when another bit is drawn. Modulation stops when all required samples are produced. The rest of the subroutine remains unchanged.

## L. SUBROUTINE MFSK

This subroutine modulates a signal simulating M-ary Frequency Shift Keying. The separation between adjacent frequencies is established as 1/Ts, where Ts is the symbol duration. This amounts to a modulation index equal to 1. Initially the entire range of frequency deviation from the carrier is calculated as (n-1)Ts. The mean frequency deviation is then calculated. In other words the range of frequencies of the modulated signal is the frequency of the central carrier ±1/2 the magnitude of the entire range of frequency deviation. The minimum frequency is used as the base for computing the frequency to be used to modulate the signal during each respective symbol duration.

At this point in the subroutine a series of k random bits ($k \leq 10$) is drawn and converted to decimal. This decimal number is multiplied by the frequency separation and added to the minimum frequency and converted to radians. The analytical expression of the signal produced with this subroutine is illustrated in equation 6.8.

$vc(t) = A \cos[(w + 2pi\ mf)t + delta]$ (eqn 6.8)

mf = modulation frequency

This modulation continues until a symbol duration elapses when a new frequency deviation or mf is calculated or modulation stops do to completion of all required samples.

## H. SUBROUTINE QPRS

This module simulates 5 different classes of Quadrature Partial Response Signalling systems. The technique employed involves the summation, at each time increment, of all responses to the linear filter representing the class. This is accomplished for all of the bits being modulated during the duration of the simulation.

Initally, two arrays of random bits are generated representing those bits which would be modulated onto the quadrature components of the carrier during the length of time the signal is to be simulated. The the impulse response to each of these n bits is calculated for the time of the respective sample. These impulse responses represent the modulation for the bit in question onto the respective porticns of the carrier and the sum is formed according to equation 6.9.

$$vc(t) = \sum_{n=-\infty}^{\infty} A\ h(t - nTs)\ \cos(wt + delta) + \sum_{n=-\infty}^{\infty} A\ h(t - nTs)\ \sin(wt + delta) \qquad \text{(eqn 6.9)}$$

h(t - nTs) = impulse response of nth bit at time t

The time of the first sample is initialized at 6(BAUDD) in order to ensure that when the first sample is taken, contributions from bits modulated at time 0 are also summed do to the overlapping nature of the QPR signals. The value of the last sample is also formed by the sum of the quadrature components existing at 6(BAUDD) after it is produced. The rest of the program operation remains the same as previous modules.

## N. SUBROUTINE STREAM

This subroutine interacts with the IMSL random number generator SUBROUTINE GGUBFS to produce a random number between 0 and 1 and make assignment on the basis of the value of this random number to random bits according to the digital logic scheme specified in the MAIN program. It also enables the user to manually insert binary digits if that option was specified previously in MAIN.

## O. SUBROUTINE ORTHO

ORTHO produces an n X n orthogonal matrix from the Hadamard matrix by forming successive Kronecker products [Ref. 15]. It also selects the appropriate row and column of the orthogonal bit to be modulated and passes this bit back to SUBROUTINE OBPSK. Point of entry to SUBROUTINE ORTHO is controlled by a flag set in OBPSK.

## P. SUBROUTINE PLOT

PLOT is an interactive module that allows the user to determine whether or not a graph of the output of program calculations is to be produced. The user is also able to selectively vary certain parameters associated with the plot. PLOT calls SUBROUTINE UTPLOT which actually produces the graph and performs the output. UTPLOT is a NON-IMSL library routine from the Naval Postgraduate School.

## Q. SUBROUTINE STATS

The statistical calculations associated with the amplitude spectrums of the various functions generated on successive repetitions of the program are computed in SUBROUTINE STATS. Upon completion of each repetition, each module calls STATS where the sum, sum of the squares, sum of the cubes and sum of the quartics of each element of the amplitude spectrum of the signal produced are computed and stored. When the last repetition of the program is complete, MAIN instructs STATS to compute and output the final statistics, i.e., mean, variance, skewness and kurtosis of each component of the amplitude spectrum according to the estimations contained in [Ref. 16]. Additionally the mean variance and variance of the variance is calculated and output. Point of entrance to this subroutine is controlled by flags set in the individual modulation subroutines or in MAIN.

# VII. STATISTICS OF THE FAST FOURIER TRANSFORMS

## A. DESCRIPTION OF THE PROBLEM

As stated in Chapter II, if it can be shown that the statistics of the FFT can be somehow linked to a particular digital signal modulation technique, then the modulation technique can be identified on the basis of those statistics alone. Although it is beyond the scope of this thesis to actually derive those relationships if they exist, it seem prudent to attempt to determine if there is statistical differences between the components of the FFT's as they are derived in the enclosed computer simulations.

The statistic chosen to do the hypothesis testing is the F-test since the true mean and variance of the distribution need not be known [Ref. 17]. In addition, the information necessary to calculate the F-statistic is readily accumulated in SUBROUTINE STATS. Calculation of the F-statistic necessitated the writing of a small computer program to be used for that purpose once the output from the main program was generated and reformatted.

## B. ANALYSIS-OF-VARIANCE

If there were no differences attributable to the modulation technique, then it would be reasonable to assume that for a certain set of characteristics, the components of successive FFT's would be the same. In other words, if a signal was modulated by BPSK and a statistically significant number of FFT's were generated of the signal, then the statistics of those FFT's would be assumed to be related. On the other hand, if it can be shown that the statistics are not related (i.e., not from the same distribution), then significance can be placed in the variation among modulation techniques.

### 1. The F-Distribution

The assumptions necessary to use the F-distribution are:

1. Normal distribution
2. Random samples
3. Independent samples

These assumptions are not too difficult to intuitively accept in the model that will be proposed. Once again it would be expected that the FFT's of successive identically modulated signals to be related. For the simulations conducted as part of this test, they were all generated using bits from a random number generator for which it can be shown that each sequence of bits passes statistical tests for randomness and independence. The assumption of normality could also be argued on the basis of the central limit theorem.

The value of the F-statistic with which comparison will be made is 1.51 for 14 degrees of freedom in the numerator, an infinite number of degrees of freedom in the denominator and a 90% confidence region. How these values were obtained will be detailed under the design of the experiment.

### 2. Design of the Experiment

The experimental data used in the computation of the statistics was chosen to minimize the random contributions of variables other than the modulation technique. The computer simulations included in this thesis were used to generate the statistics and a separate program also included in the appendix was used to calculate the F-statistic. The variables in the experiment include:

1. Modulation technique
2. Logic type
3. Baud rate
4. Bits per binary code word
5. Carrier frequency
6. Carrier amplitude
7. Initial phase angle
8. Number of samples generated
9. Time between samples
10. Seed for random number generator
11. Number of repetitions or trials

Fifteen different modulation techniques were compared. In all cases bipolar logic was simulated, the baud rate was held constant at 1200 baud and the maximum carrier amplitude was established at 1 volt. Additionally the phase angle of all simulations was 0 degrees and the seed for the random number generator was 1 thereby ensuring the same random sequence of bits. The number of bits per binary code word did vary among the modulation techniques. When the modulation technique was M-ary, the bits per code word was always 3 so it is possible to infer that bits per code word is a function of the modulation technique. The time between samples was the normalized Nyquist sampling rate. This did vary between modulation techniques but was necessary to derive an accurate FFT. Carrier frequency also varied between simulations but was always twice the lowest carrier frequency recommended in the computer simulation. In most cases this was 2400 Hz, or twice the baud rate. Finally each simulation was repeated 100 times and the statistics of the FFT's based on those 100 repetitions. The number of samples produced was always 64 so there are 33 components of the FFT.

What this amounts to can be illustrated through an Analysis-of-Variance table as shown in Table 2 [Ref. 17].

## TABLE 2
## ANALYSIS-OF-VARIANCE TABLE

$n$ Observations in Each of $r$ Groups

| | | Observations | Sum | Mean |
|---|---|---|---|---|
| Group | 1 | $Y_{11}$ $Y_{12}$ $Y_{13}$ ...... $Y_{1n}$ | $Y_{1.}$ | $\overline{Y}_{1.}$ |
| | 2 | $Y_{21}$ $Y_{22}$ $Y_{23}$ ...... $Y_{2n}$ | $Y_{2.}$ | $\overline{Y}_{2.}$ |
| | 3 | $Y_{31}$ $Y_{32}$ $Y_{33}$ ...... $Y_{3n}$ | $Y_{3.}$ | $\overline{Y}_{3.}$ |
| | ⋮ | | | |
| | $r$ | $Y_{r1}$ $Y_{r2}$ $Y_{r3}$ ...... $Y_{rn}$ | $Y_{r.}$ | $\overline{Y}_{r.}$ |

In the case of the simulation described above, this amounts to 100 observations (repetitions or trials) from each of 15 groups (modulation techniques) for each of the 33 components of the FFT. The degrees of freedom in the numerator is equal to (15-1) or 14 while the degrees of freedcm in the denominator is equal to 15(100-1) or 1485. Table values for the F-distribution use infinity as the degree of freedom for values greater than 120.

An F-test was also performed to determine if there is a relationship among the mean variance, mean skewness or mean kurtosis of the 15 different modulation techniques. Again the assumption is made that these statistics would be normally distributed from modulation techniques which had the same or statistically similar components of their FFT's. In this case there were again 15 modulation techniques to compare (14 degrees of freedom) but only 33 observations (the variance, skewness or kurtosis of the respective elements of the FFT). This number of observations yields 15(33-1) = 480 degrees of freedom in the denominator.

### 3. Hypothesis and Hypothesis Testing

The hypothesis posed for the model is that the means of the individual components of the FFT's from the 15 modulation techniques are from the same distribution. Therefore, the respective means must be equal. Also tested is that the mean variance, mean skewness and mean kurtosis of all components of the FFT's of each modulation technique are equal. These hypothesis are illustrated in equations 7.1 through equation 7.4. Each hypothesis is tested and compared to the F-distribution selecting a rejection region or level of significance of .9. This equates to an F-statistic of 1.51 for 14 degrees of freedom in the numerator and an infinite number of degrees of freedom in the denominator.

### 4. Results

The means of the 33 components of the FFT for each of the 15 modulation techniques were compared using the F-distribution. The results are shown in Table 3. Table 4 shows the F-statistics associated with the comparison of the mean of the variance, mean of the skewness and mean of the kurtosis for the respective groups. In addition a complete summation of the results of the F-test are included in the appendix.

## C. CONCLUSION

The results of the F-test indicate that the null hypothesis must be rejected at the .9 significance level and the alternate hypothesis accepted that the means are not equal for FFT components 9-14. In addition the F-statistic for the comparison of mean variance, mean skewness and mean kurtosis all fall in the rejection area.

$$Ho: XBAR11 = XBAR21 = \bullet\bullet\bullet = XBARij \qquad \text{(eqn 7.1)}$$

$$H1: XBAR11 \neq XBAR21 \neq \bullet\bullet\bullet \neq XBARij$$

i = ith modulation technique
j = jth component of the FFT

$$Ho: MVAR1 = MVAR2 = \bullet\bullet\bullet = MVAR15 \qquad \text{(eqn 7.2)}$$

$$H1: MVAR1 \neq MVAR2 \neq \bullet\bullet\bullet \neq MVAR\ 15$$

$$H0: MSKEW1 = MSKEW2 = \bullet\bullet\bullet = MSKEW15 \qquad \text{(eqn 7.3)}$$

$$H1: MSKEW1 \neq MSKEW2 \neq \bullet\bullet\bullet \neq MSKEW15$$

$$Ho: MKUR1 = MKUR2 = \bullet\bullet\bullet = MKUR15 \qquad \text{(eqn 7.4)}$$

$$H1: MKUR1 \neq MKUR2 \neq \bullet\bullet\bullet \neq MKUR15$$

This indicates that there is a difference between the statistics of of the FFT's of the respective modulation technique. Since all the parameters used in the generation of these statistics were held essentially constant among the trials, it can be assumed that these differences are due to the modulation technique employed. It has not been established yet what those differences may be. This is an area where future research and study is warranted.

The results of this test would seem to point to those components of the FFT which offer the best opportunity to develop those relationships or differences. FFT components 9-14 have obvious differences; however, FFT components 20, 24 and 25 also have large F-statistics but do not fall in the rejection area. These components may also be

TABLE 3

F-STATISTICS OF FFT COMPONENTS

| FFT COMPONENT | AMONG GROUP SUM | WITHIN GROUP SUM | TOTAL OF SQUARES | F-STAT |
|---|---|---|---|---|
| 1 | 0.121 | 17.69 | 17.81 | 0.727 |
| 2 | 0.112 | 18.48 | 18.59 | 0.642 |
| 3 | 0.108 | 20.65 | 20.76 | 0.557 |
| 4 | 0.135 | 25.40 | 25.54 | 0.562 |
| 5 | 0.172 | 32.74 | 32.91 | 0.558 |
| 6 | 0.254 | 41.72 | 41.98 | 0.647 |
| 7 | 0.422 | 50.10 | 50.52 | 0.893 |
| 8 | 0.962 | 82.83 | 83.79 | 1.231 |
| 9 | 7.591 | 390.2 | 397.8 | 2.063 |
| 10 | 49.09 | 1681.0 | 1730.1 | 3.097 |
| 11 | 53.01 | 1732.1 | 1785.1 | 3.246 |
| 12 | 61.56 | 2071.7 | 2133.3 | 3.152 |
| 13 | 45.73 | 1832.8 | 1878.6 | 2.646 |
| 14 | 3.864 | 250.5 | 254.4 | 1.636 |
| 15 | 0.539 | 98.13 | 98.67 | 0.582 |
| 16 | 0.366 | 110.1 | 110.5 | 0.353 |
| 17 | 0.504 | 151.0 | 151.5 | 0.354 |
| 18 | 0.943 | 140.3 | 141.2 | 0.713 |
| 19 | 1.299 | 158.5 | 159.8 | 0.870 |
| 20 | 2.671 | 200.7 | 203.4 | 1.412 |
| 21 | 1.853 | 226.3 | 228.2 | 0.868 |
| 22 | 2.277 | 232.1 | 234.3 | 1.041 |
| 23 | 1.712 | 197.6 | 199.3 | 0.919 |
| 24 | 2.512 | 181.9 | 184.4 | 1.476 |
| 25 | 2.144 | 171.3 | 173.5 | 1.327 |
| 26 | 1.257 | 143.7 | 145.0 | 0.928 |
| 27 | 1.233 | 113.9 | 115.1 | 1.149 |

TABLE 3
F-STATISTICS OF FFT COMPONENTS (con't)

| | | | | |
|---|---|---|---|---|
| 28 | 0.858 | 115.8 | 116.7 | 0.785 |
| 29 | 0.547 | 63.72 | 64.27 | 0.911 |
| 30 | 0.356 | 55.77 | 56.13 | 0.677 |
| 31 | 0.183 | 41.41 | 51.59 | 0.469 |
| 32 | 0.051 | 13.79 | 13.84 | 0.389 |
| 33 | 0.031 | 10.92 | 10.96 | 0.304 |

**TABLE 4**

**F-STATISTICS OF THE MEAN VARIANCES, MEAN SKEWNESS AND MEAN KURTOSIS**

| | AMONG GROUP SUM | WITHIN GROUP SUM | TOTAL OF SQUARES | F-STAT |
|---|---|---|---|---|
| VARIANCE | 3.144 E5 | 3.755 E6 | 4.070 E6 | 2.870 |
| SKEWNESS | 5.307 E10 | 7.031 E11 | 7.562 E11 | 2.588 |
| KURTOSIS | 8.280 E11 | 1.262 E13 | 1.344 E13 | 2.250 |

interesting to examine. In addition, it should be noted that the statistics of the FFT associated with each respective modulation technique also display some striking differences. These statistics may prove in some way to be a fingerprint of the modulation techniques themselves.

# APPENDIX A

## FORTRAN PROGRAM FOR DIGITAL SIGNAL MODULATION

```
C     ****************************************************************MAI00020
C     *                                                              *MAI00030
C     * TITLE:  DIGITAL SIGNAL MODULATION TECHNIQUES                  *MAI00040
C     *                                                              *MAI00050
C     * PURPOSE:  THIS PROGRAM GENERATES A SIMULATED WAVEFORM WHICH IS *MAI00060
C     * CHARACTERISTIC OF ONE OF A VARIETY OF DIGITAL SIGNAL MODULATION*MAI00070
C     * TECHNIQUES.  THE PROGRAM IS INTERACTIVE AND DESIGNED TO BE AS  *MAI00080
C     * USER FRIENDLY AS POSSIBLE.  THE PROGRAM ALLOWS THE USER TO     *MAI00090
C     * SPECIFY ANY ONE OF A NUMBER OF PARAMETERS OR VARIABLES WHICH   *MAI00100
C     * ARE CHARACTERISTIC OF THE INDIVIDUAL MODULATION TECHNIQUE.     *MAI00110
C     * THE PROGRAM ALSO ALLOWS THE USER TO DEVELOP A PLOT OF THE      *MAI00120
C     * SIMULATED WAVEFORM, GENERATE THE FAST FOURIER TRANSFORM,       *MAI00130
C     * OR DO LIMITED STATISTICAL ANALYSIS.                            *MAI00140
C     *                                                              *MAI00150
C     * WRITTEN BY:  CRAIG CARLSON, LCDR, USN                          *MAI00160
C     * IN PARTIAL FULFILLMENT OF MASTER OF SCIENCE, SYSTEMS TECHNOLOGY*MAI00170
C     *                                                              *MAI00180
C     * THIS PROGRAM WAS WRITTEN FOR EXECUTION ON AN IBM 3033 COMPUTER *MAI00190
C     * UTILIZING THE VMS/CMS OPERATING SYSTEM.  WITHIN THE CMS SYSTEM *MAI00200
C     * THERE IS A ROUTINE TO CLEAR THE SCREEN WHEN DESIRED.  THIS     *MAI00210
C     * FUNCTION IS CALLED THROUGH USE OF THE COMMAND *                 MAI00220
C     * CALL FRTCMS('CLRSCRN ').  IF THE PROGRAM IS TO BEUSED ON THIS* MAI00230
C     * OPERATING SYSTEM, REMOVAL OF THE COMMENT C BEFORE LINES OF     *MAI00240
C     * CODE CONTAINING THIS COMMAND WILL IMPROVE THE INTERACTIVE      *MAI00250
C     * PORTION OF THE PROGRAM.*                                        MAI00260
C     *                                                              *MAI00270
C     ****************************************************************MAI00280
C                                                                       MAI00290
C     *** MAIN CONTROL MODULE ***                                       MAI00300
C                                                                       MAI00310
C     *** PURPOSE ***                                                   MAI00320
C                                                                       MAI00330
C     THIS PROGRAM IS THE MAIN BODY OF THE SIGNAL GENERATOR.  IT        MAI00340
C     CONTROLS INPUT AND OUTPUT AND CALLS VARIOUS SUBROUTINES DURING    MAI00350
C     ITS OPERATION.                                                    MAI00360
C                                                                       MAI00370
C     *** VARIABLE DEFINITIONS ***                                      MAI00380
C                                                                       MAI00390
C     ANS=         A INTEGER REPRESENTING THE RESPONSE TO AN INQUIRY    MAI00400
C     TYPE1=       AN INTEGER TO CONTROL INPUT                          MAI00410
C     TYPE2=       AN INTEGER TO CONTROL INPUT                          MAI00420
C     TYPE3=       AN INTEGER TO CONTROL INPUT                          MAI00430
C     ANS2=        AN INTEGER USED TO SPECIFY HOW BIT STREAM IS TO BE   MAI00440
```

```
C                 GENERATED
C     IBAUD=      INTEGER VALUE OF BAUD RATE
C     BAUD=       SYMBOL RATE OR BAUD RATE
C     BAUDD=      BAUD DURATION
C     IBITS=      INTEGER VALUE OF BITS IN A BINARY CODE WORD
C     BITS=       NUMBER OF BITS IN EACH BINARY CODE WORD
C     BITR=       BIT RATE
C     IFREQ=      INTEGER VALUE OF CARRIER FREQUENCY
C     FREQ=       CARRIER FREQUENCY
C     IIPHAS=     INTEGER VALUE OF INITIAL PHASE ANGLE IN DEGREES
C     IPHAS=      INITIAL PHASE ANGLE IN DEGREES
C     IAMP=       INTEGER VALUE OF IN-PHASE AMPLITUDE
C     AMP=        AMPLITUDE
C     ISEED=      INTEGER VALUE OF SEED FOR RANDOM NUMBER GENERATOR
C     DSEED=      SEED FOR RANDOM NUMBER GENERATOR
C     ARRAY=      A TWO DIMENTIONAL ARRAY
C     R=          AN INTEGER REPRESENTING THE ROW OF ARRAY
C     IN=         INTEGER OF NUMBER OF POSITIONS TO BE USED IN ARRAY
C     N=          REAL VALUE OF IN
C     REPS=       INTEGER VALUE OF NUMBER OF TIMES THE PROGRAM IS TO BE
C                 REPEATED
C     REP=        INTEGER VALUE OF THE NUMBER OF THE REPETITION OF THE
C                 PROGRAM AND PASSED TO THE SUBPROGRAM FOR THE
C                 MODULATION TECHNIQUE BEING EMPLOYED
C     FLAG=       INTEGER TO CONTROL POINT OF ENTRANCE TO SUBROUTINE
C                 STATS
C     FC=         RECOMMENDED MINIMUM CARRIER FREQUENCY
C     STEP=       TIME INTERVAL BETWEEN SAMPLES
C
C     *** VARIABLE DECLARATIONS ***
C
      INTEGER ANS,TYPE1,TYPE2,TYPE3,ANS2,IBAUD,IBITS,IFREQ,
     *IIPHAS,IAMP,ISEED,R,IN,REPS,REP,FLAG
C
      DOUBLE PRECISION BAUD,BITS,BITR,FREQ,IPHAS,AMP,BAUDD,STEP,FC,
     *DSEED,N,PI
C
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
     *SUMX4(513)
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
C     *** VARIABLE INITIALIZATIONS ***
C
      R=1
      PI=3.141592653589793D0
      TYPE3=0
C
C     *** INTRODUCE THE PROGRAM ***
C
C     CALL FRTCMS('CLRSCRN ')
```

```
C
10    WRITE(10,20)
20    FORMAT(' HELLO!  YOU HAVE INITIATED A PROGRAM WHICH WILL ALLOW',/,
     *' YOU TO PRODUCE ONE OF A NUMBER OF SIMULATED COMMUNICATIONS',/,
     *' SIGNALS UTILIZING VARIOUS DIGITAL SIGNAL MODULATION TECHNIQUES.'
     *,/,' YOU WILI BE ASKED TO PROVIDE CERTAIN INFORMATION ABOUT THE',
     */,' SIGNAL YOU WISH TO SIMULATE.  ADDITIONALLY YOU WILL BE',
     */,' GIVEN THE OPPORTUNITY TO PLOT THE RESULTS, GENERATE THE FFT'
     *,/,' OF THE SIGNAL OR TO DO CERTAIN STATISTICAL ANALYSIS OF THE'
     *,/,' FFT.  IF YOU ARE READY TO PROCEED, TYPE 1 <CR>.',//,' :')
C
      READ(5,21)ANS
21    FORMAT(I1)
C
      IF(ANS.NE.1)THEN
C        CALL FRTCMS('CLRSCRN ')
         GO TO 10
      END IF
C
C     *** DISPLAY MODULATION TECHNIQUES AVAILABLE AND INQUIRE WHICH
C     TECHNIQUE IS TO BE SIMULATED ***
C
C     CALL FRTCMS('CLRSCRN ')
C
29    WRITE(10,30)
30    FORMAT(' THIS PROGRAM IS CAPABLE OF SIMULATING THE FOLLOWING',/,
     *' DIGITAL SIGNAL MODULATION TECHNIQUES:',//,
     *' 1.  BINARY PHASE SHIFT KEYING (BPSK)',/,
     *' 2.  DIFFERENTIAL BINARY PHASE SHIFT KEYING (DBPSK)',/,
     *' 3.  ORTHOGONAL BINARY PHASE SHIFT KEYING (ORTHOGONAL BPSK)',/,
     *' 4.  QUADRATURE PHASE SHIFT KEYING (QPSK)',/,
     *' 5.  OFFSET QUADRATURE PHASE SHIFT KEYING (OQPSK)',/,
     *' 6.  MULTIPLE PHASE SHIFT KEYING (MPSK)',/,
     *' 7.  MULTIPIE AMPLITUDE SHIFT KEYING (MASK)',/,
     *' 8.  QUADRATURE AMPLITUDE SHIFT KEYING (QASK)',/,
     *' 9.  MINIMUM SHIFT KEYING (MSK) OR',/,
     *'     FAST FREQUENCY SHIFT KEYING (FFSK)',/,
     *' 10. MULTIPLE FREQUENCY SHIFT KEYING (MFSK)',/,
     *' 11. QUADRATURE PARTIAL RESPONSE (QPRS)',//,
     *' ENTER THE NUMBER (1-11) WHICH CORRESPONDS TO THE DIGITAL SIGNAL'
     *,/,' SIGNAL MODULATION TECHNIQUE YOU WISH TO SIMULATE',//,' :')
C
      READ(5,*)TYPE1
C
      IF(TYPE1.LT.1.OR.TYPE1.GT.11)THEN
C        CALL FRTCMS('CLRSCRN ')
         WRITE(10,40)
40       FORMAT(' ERROR',/)
         GO TO 29
      END IF
```

```
C
C     *** DETERMINE THE CLASS OF THE QUADRATURE PARTIAL RESPONE SYSTEM *
C
      IF(TYPE1.EQ.11) THEN
C
          CALL FRTCMS('CLRSCRN ')
C
41        WRITE(10,42)
42        FORMAT(' ENTER THE CLASS OF THE QUADRATURE PARTIAL RESPONSE',
     *    /,' SYSTEM TO BE SIMULATED FROM 1-5.',//,' :')
C
          READ(5,*)TYPE3
C
          IF(TYPE3.LT.1.OR.TYPE3.GT.5) THEN
              CALL FRTCMS('CLRSCRN ')
C
              WRITE(10,43)
43            FORMAT(' ERROR',/)
              GO TO 41
          END IF
C
      END IF
C
C     *** DETERMINE THE LOGIC SCHEME TO BE EMPLOYED DURING MODULATION **
C
      CALL FRTCMS('CLRSCRN ')
C
49    WRITE(10,50)
50    FORMAT(' ENTER THE NUMBER (1-3) WHICH CORRESPONDS TO THE',/,
     *' DESIRED DIGITAL LOGIC SCHEME:',//,
     *' 1.  BIPOLAR',/,
     *' 2.  UNIPOLAR POSITIVE',/,
     *' 3.  UNIPOLAR NEGATIVE',//,' :')
C
      READ(5,51)TYPE2
51    FORMAT(I1)
C
      IF(TYPE2.LT.1.OR.TYPE2.GT.3) THEN
          CALL FRTCMS('CLRSCRN ')
C
          WRITE(10,60)
60        FORMAT(' ERROR',/)
          GO TO 49
      END IF
C
C     *** HAVE THE USER ENTER THE BAUD OR SYMBOL RATE ***
C
      CALL FRTCMS('CLRSCRN ')
C
69    WRITE(10,70)
70    FORMAT(' ENTER THE BAUD RATE OR SYMBOL RATE AT THIS TIME.',//,
     *' :')
```

```
C                                                                          MAI01950
      READ(5,*)IBAUD                                                       MAI01960
C                                                                          MAI01970
      BAUD=IBAUD                                                           MAI01980
C                                                                          MAI01990
      BAUDD=1.D0/BAUD                                                      MAI02000
C                                                                          MAI02010
C     *** DETERMINE THE BITS EITHER BY THE TYPE OF DIGITAL                 MAI02020
C     MODULATION SPECIFIED OR BY USER INPUT ***                            MAI02030
C                                                                          MAI02040
      CALL FRTCMS('CLRSCRN ')                                              MAI02050
C                                                                          MAI02060
      IF(TYPE1.EQ.1.OR.TYPE1.EQ.2.OR.TYPE1.EQ.9) THEN                      MAI02070
          IBITS=1                                                          MAI02080
          BITS=IBITS                                                       MAI02090
C                                                                          MAI02100
      ELSE IF(TYPE1.EQ.3)THEN                                              MAI02110
79        WRITE(10,80)                                                     MAI02120
80        FORMAT(' ENTER THE NUMBER OF BITS IN EACH SYMBOL.',/,            MAI02130
     *    ' CAUTION!  THE NUMBER OF BITS PER SYMBOL MUST BE 6 OR LESS'     MAI02140
     *    ,//,' :')                                                        MAI02150
C                                                                          MAI02160
          READ(5,*)IBITS                                                   MAI02170
C                                                                          MAI02180
          IF(IBITS.LT.1.OR.IBITS.GT.6)THEN                                 MAI02190
              CALL FRTCMS('CLRSCRN ')                                      MAI02200
              WRITE(10,90)                                                 MAI02210
90            FORMAT(' ERROR',/)                                           MAI02220
              GO TO 79                                                     MAI02230
          ELSE                                                             MAI02240
              BITS=IBITS                                                   MAI02250
          END IF                                                           MAI02260
C                                                                          MAI02270
      ELSE IF(TYPE1.EQ.4.OR.TYPE1.EQ.5.OR.TYPE1.EQ.11)THEN                 MAI02280
          IBITS=2                                                          MAI02290
          BITS=IBITS                                                       MAI02300
C                                                                          MAI02310
      ELSE IF(TYPE1.GE.6.OR.TYPE1.LE.8.OR.TYPE1.EQ.10)THEN                 MAI02320
99        WRITE(10,100)                                                    MAI02330
100       FORMAT(' ENTER THE NUMBER OF BITS IN EACH SYMBOL.',/,            MAI02340
     *    ' CAUTION!  THE NUMBER OF BITS PER SYMBOL MUST BE 10 OR LESS'    MAI02350
     *    ,//,' :')                                                        MAI02360
C                                                                          MAI02370
          READ(5,*)IBITS                                                   MAI02380
C                                                                          MAI02390
          IF(IBITS.LT.1.OR.IBITS.GT.11)THEN                                MAI02400
              CALL FRTCMS('CLRSCRN ')                                      MAI02410
              WRITE(10,110)                                                MAI02420
110           FORMAT(' ERROR',/)                                           MAI02430
              GO TO 99                                                     MAI02440
```

```
            ELSE
                BITS=IBITS
            END IF
C
        END IF
C
C       *** DETERMINE AND DISPLAY THE BIT RATE ***
C
        IF(TYPE1.EQ.3)THEN
            BITR=(2.D0**IBITS)*BAUD
        ELSE
            BITR=BITS*BAUD
        END IF
C
C       CALL FRTCMS('CLRSCRN ')
        WRITE(10,120)BITR
120     FORMAT(' THE BIT RATE FOR THE SPECIFIED SIGNAL IS',F9.0,' BITS/SEC
       *')
C
C       *** HAVE THE USER ENTER THE CARRIER FREQUENCY ***
C
        IF(TYPE1.EQ.3) THEN
            FC=BITR
        ELSE IF(TYPE1.EQ.9) THEN
            FC=1.25D0*BAUD
        ELSE IF(TYPE1.EQ.10)THEN
            FC=BAUD+(((2.D0**IBITS)-1.D0)*(BAUD/2.D0))
        ELSE
            FC=BAUD
        END IF
C
129     WRITE(10,130)FC
130     FORMAT(' ENTER THE CARRIER FREQUENCY AT THIS TIME.',//,
       *' NOTE:  CARRIER FREQ SHOULD BE GREATER THAN',F9.0,' HZ.',//,
       *' :')
C
        READ(5,*)IFREQ
C
        FREQ=IFREQ
C
        IF(FREQ.LT.FC)THEN
C           CALL FRTCMS('CLRSCRN ')
            WRITE(10,140)
140         FORMAT(' THE CARRIER FREQUENCY IS LESS THAN RECOMMENDED.',/)
            GO TO 149
        END IF
C
C       *** HAVE THE USER ENTER THE NUMBER OF SAMPLES TO BE GENERATED ***
C
C       CALL FRTCMS('CLRSCRN ')
```

```
C
149    WRITE(10,150)
150    FORMAT(' ENTER THE NUMBER OF SAMPLES OF THE MODULATED SIGNAL',
      */,' TO BE PRODUCED.  THIS NUMBER MUST BE EQUAL TO 2**N,',/,
      *' WHERE N IS A POSITIVE INTEGER EQUAL TO OR LESS THAN 10.',/,
      *' I.E., 2, 4, 8, ... , 1024',//,' :')
C
       READ(5,*)IN
       N=IN
C
       IF(IN.EQ.2.OR.IN.EQ.4.OR.IN.EQ.8.OR.IN.EQ.16.OR.IN.EQ.32.OR.
      *IN.EQ.64.OR.IN.EQ.128.OR.IN.EQ.256.OR.IN.EQ.512.OR.IN.EQ.1024)THEN
          GO TO 170
       ELSE
C         CALL FRTCMS('CLRSCRN ')
          WRITE(10,160)
160       FORMAT(' ERROR',/)
          GO TO 149
       END IF
C
C      *** DETERMINE AND DISPLAY INFO ABOUT THE RECORD LENGTH ***
C
170    IF(TYPE1.EQ.2)THEN
          STEP=1.D0/(2.D0*(FREQ+BITR))
       ELSE IF(TYPE1.EQ.9)THEN
          STEP=1.D0/(2.D0*(FREQ+(1.25D0*BAUD)))
       ELSE IF(TYPE1.EQ.10)THEN
          STEP=1.D0/(2.D0*(FREQ+(((2.D0**IBITS)-1.)*(BAUD/2.D0))+BAUD))
       ELSE IF(TYPE1.EQ.11)THEN
          STEP=1.D0/(2.D0*(FREQ+((PI+1.D0)*BAUD)))
       ELSE
          STEP=1.D0/(2.D0*(FREQ+BAUD))
       END IF
C
       IF(TYPE1.EQ.3.AND.((N-1.)*STEP).LE.1./BITR)THEN
C         CALL FRTCMS('CLRSCRN ')
          WRITE(10,180)
180       FORMAT(' ELAPSED TIME FOR SIGNAL GENERATION WILL BE LESS THAN'
      *   ,/,' 1 BIT DURATION.  IF YOU WISH TO CHANGE THIS BY',/,
      *   ' INCREASING THE NUMBER OF SAMPLES TO BE PRODUCED, BY',/,
      *   ' DECREASING CARRIER FREQUENCY, BY DECREASING THE BAUD',/,
      *   ' RATE OR BY DECREASING THE SIZE OF THE BINARY CODE WORD,',/,
      *   ' ENTER A 1.  ENTER ANY OTHER INTEGER VALUE TO CONTINUE',/
      *   ' WITH THE SIMULATION AS SPECIFIED.',//,' :')
C
          READ(5,*)ANS
C
          IF(ANS.EQ.1)THEN
             GO TO 29
          END IF
```

```
C
      ELSE IF(((N-1.D0)*STEP).LE.BAUDD)THEN
C         CALL FRTCMS('CLRSCRN ')
          WRITE(10,190)
190       FORMAT(' ELAPSED TIME FOR SIGNAL GENERATION WILL BE LESS THAN'
     *    ,/,' 1 SYMBOL DURATION.  IF YOU WISH TO CHANGE THIS BY',/,
     *    ' INCREASING THE NUMBER OF SAMPLES TO BE PRODUCED, BY',/,
     *    ' DECREASING CARRIER FREQUENCY, BY DECREASING THE BAUD',/,
     *    ' RATE OR BY DECREASING THE SIZE OF THE BINARY CODE WORD,',/,
     *    ' ENTER A 1.  ENTER ANY OTHER INTEGER VALUE TO CONTINUE',/
     *    ' WITH THE SIMULATION AS SPECIFIED.',//,' :')
C
          READ(5,*)ANS
C
          IF(ANS.EQ.1)THEN
              GO TO 29
          END IF
C
      END IF
C
C     *** HAVE THE USER ENTER THE AMPLITUDE OF THE CARRIER WAVE ***
C
C     CALL FRTCMS('CLRSCRN ')
C
199   WRITE(10,200)
200   FORMAT(' ENTER THE AMPLITUDE OF THE CARRIER.',//,' :')
C
      READ(5,*)IAMP
C
      AMP=IAMP
C
C     *** HAVE THE USER ENTER THE INITIAL PHASE ANGLE ***
C
C     CALL FRTCMS('CLRSCRN ')
C
209   WRITE(10,210)
210   FORMAT(' ENTER THE INITIAL PHASE ANGLE FROM 0 TO 360 DEGREES.',//,
     *' :')
C
      READ(5,*)IIPHAS
C
      IF(IIPHAS.LT.0.OR.IIPHAS.GT.360)THEN
C         CALL FRTCMS('CLRSCRN ')
          WRITE(10,220)
220       FORMAT(' ERROR',/)
          GO TO 209
      ELSE
          IPHAS=IIPHAS
      END IF
C
```

```
C      *** DETERMINE IF A RANDOM BIT STREAM IS TO BE GENERATED ***
C
C      CALL FRTCMS('CLRSCRN ')
C
229    WRITE(10,230)
230    FORMAT(' ENTER 1 IF YOU DESIRE TO HAVE THE COMPUTER GENERATE A',/,
     *' RANDOM BIT STREAM OR ENTER 2 IF YOU WISH TO ENTER YOUR OWN BIT',
     */,' STREAM.',//,' :')
C
       READ(5,240) ANS2
240    FORMAT(I1)
C
       IF(ANS2.LT.1.OR.ANS2.GT.2) THEN
C          CALL FRTCMS('CLRSCRN ')
           WRITE(10,250)
250        FORMAT(' ERROR',/)
           GO TO 229
       END IF
C
C      *** HAVE THE USER ENTER THE SEED FOR THE RANDOM NUMBER GENERATOR**
C
C      CALL FRTCMS('CLRSCRN ')
C
       IF(ANS2.EQ.1)THEN
           WRITE(10,260)
260        FORMAT(' ENTER THE SEED (A REAL NUMBER BETWEEN 1 AND 2**31)',/
     *     ,' FOR THE RANDOM NUMBER GENERATOR',//,' :')
C
           READ(5,*)ISEED
C
           DSEED=ISEED
       END IF
C
C      *** HAVE THE USER SPECIFY THE NUMBER OF TRIALS TO BE USED ***
C
C      CALL FRTCMS('CLRSCRN ')
C
       WRITE(10,270)
270    FORMAT(' ENTER THE NUMBER OF TIMES YOU WOULD LIKE THE SIMULATION',
     */,' TO REPEAT.',//,' :')
C
       READ(5,*)REPS
C
C      *** CALL THE APPROPRIATE SUBROUTINE FOR THE MODULATION TECHNIQUE
C      SPECIFIED, THE APPROPRIATE NUMBER OF TIMES ***
C
       DO 280 REP=1,REPS
C
       IF(TYPE1.EQ.1)THEN
           CALL BPSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)
```

```
C
      ELSE IF(TYPE1.EQ.2)THEN
          CALL DBPSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)
C
      ELSE IF(TYPE1.EQ.3)THEN
          CALL OBPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)
C
      ELSE IF(TYPE1.EQ.4)THEN
          CALL QPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)
C
      ELSE IF(TYPE1.EQ.5)THEN
          CALL OQPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,
     *    REP)
C
      ELSE IF(TYPE1.EQ.6)THEN
          CALL MPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,
     *    REP)
C
      ELSE IF(TYPE1.EQ.7)THEN
          CALL MASK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,
     *    REP)
C
      ELSE IF(TYPE1.EQ.8)THEN
          CALL QASK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,
     *    REP)
C
      ELSE IF(TYPE1.EQ.9)THEN
          CALL MSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)
C
      ELSE IF(TYPE1.EQ.10)THEN
          CALL MFSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)
C
      ELSE IF(TYPE1.EQ.11)THEN
          CALL QPRS(TYPE2,TYPE3,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,
     *    IN,REP)
C
      END IF
C
280   CONTINUE
C
C     *** OUTPUT A FINGERPRINT OF THE MODULATION ACCOMPLISHED ***
C
      IF(TYPE1.EQ.1)THEN
          WRITE(6,281)
281       FORMAT('1',' MODULATION TECHNIQUE = BPSK')
      ELSE IF(TYPE1.EQ.2)THEN
          WRITE(6,282)
282       FORMAT('1',' MODULATION TECHNIQUE = DBPSK')
      ELSE IF(TYPE1.EQ.3)THEN
          WRITE(6,283)
```

```
283      FORMAT('1',' MODULATION TECHNIQUE = ORTHOGONAL BPSK')
      ELSE IF(TYPE1.EQ.4)THEN
         WRITE(6,284)
284      FORMAT('1',' MODULATION TECHNIQUE = QPSK')
      ELSE IF(TYPE1.EQ.5)THEN
         WRITE(6,285)
285      FORMAT('1',' MODULATION TECHNIQUE = OQPSK')
      ELSE IF(TYPE1.EQ.6)THEN
         WRITE(6,286)
286      FORMAT('1',' MODULATION TECHNIQUE = MPSK')
      ELSE IF(TYPE1.EQ.7)THEN
         WRITE(6,287)
287      FORMAT('1',' MODULATION TECHNIQUE = MASK')
      ELSE IF(TYPE1.EQ.8)THEN
         WRITE(6,288)
288      FORMAT('1',' MODULATION TECHNIQUE = QASK')
      ELSE IF(TYPE1.EQ.9)THEN
         WRITE(6,289)
289      FORMAT('1',' MODULATION TECHNIQUE = MSK')
      ELSE IF(TYPE1.EQ.10)THEN
         WRITE(6,290)
290      FORMAT('1',' MODULATION TECHNIQUE = MFSK')
      ELSE IF(TYPE1.EQ.11)THEN
         WRITE(6,291)
291      FORMAT('1',' MODULATION TECHNIQUE = QPRS')
      END IF
C
      IF(TYPE1.EQ.11.AND.TYPE3.EQ.1)THEN
         WRITE(6,292)
292      FORMAT(' QPRS CLASS 1 FILTER')
      ELSE IF(TYPE1.EQ.11.AND.TYPE3.EQ.2)THEN
         WRITE(6,293)
293      FORMAT(' QPRS CLASS 2 FILTER')
      ELSE IF(TYPE1.EQ.11.AND.TYPE3.EQ.3)THEN
         WRITE(6,294)
294      FORMAT(' QPRS CLASS 3 FILTER')
      ELSE IF(TYPE1.EQ.11.AND.TYPE3.EQ.4)THEN
         WRITE(6,295)
295      FORMAT(' QPRS CLASS 4 FILTER')
      ELSE IF(TYPE1.EQ.11.AND.TYPE3.EQ.5)THEN
         WRITE(6,296)
296      FORMAT(' QPRS CLASS 5 FILTER')
      END IF
C
      IF(TYPE2.EQ.1)THEN
         WRITE(6,297)
297      FORMAT(' BIPOLAR LOGIC')
      ELSE IF(TYPE2.EQ.2)THEN
         WRITE(6,298)
298      FORMAT(' UNIPOLAR POSITIVE LOGIC')
```

```
      ELSE IF(TYPE2.EQ.3)THEN                                            MAI05450
          WRITE(6,299)                                                   MAI05460
299       FORMAT(' UNIPOLAR NEGATIVE LOGIC')                             MAI05470
      END IF                                                             MAI05480
C                                                                        MAI05490
      WRITE(6,300)IBAUD                                                  MAI05500
300   FORMAT(' BAUD OR SYMBOL RATE = ',I10,' HZ')                        MAI05510
C                                                                        MAI05520
      WRITE(6,301)IBITS                                                  MAI05530
301   FORMAT(' BITS PER BINARY CODE WORD = ',I10)                        MAI05540
C                                                                        MAI05550
      WRITE(6,302)BITR                                                   MAI05560
302   FORMAT(' BIT RATE = ',E23.16)                                      MAI05570
C                                                                        MAI05580
      WRITE(6,303)IFREQ                                                  MAI05590
303   FORMAT(' CARRIER FREQUENCY = ',I10,' HZ')                          MAI05600
C                                                                        MAI05610
      WRITE(6,304)IAMP                                                   MAI05620
304   FORMAT(' MAXIMUM CARRIER AMPLITUDE = ',I10,' VOLT(S)')             MAI05630
C                                                                        MAI05640
      WRITE(6,305)IIPHAS                                                 MAI05650
305   FORMAT(' INITIAL PHASE ANGLE = ',I10,' DEGREES')                   MAI05660
C                                                                        MAI05670
      WRITE(6,306)STEP                                                   MAI05680
306   FORMAT(' TIME BETWEEN SAMPLES = ',E23.16,' SEC')                   MAI05690
C                                                                        MAI05700
      WRITE(6,307)IN                                                     MAI05710
307   FORMAT(' NUMBER OF SAMPLES GENERATED = ',I10)                      MAI05720
C                                                                        MAI05730
      IF(ANS2.EQ.1)THEN                                                  MAI05740
          WRITE(6,308)ISEED                                              MAI05750
308       FORMAT(' SEED FOR RANDOM NUMBER GENERATOR = ',I10)             MAI05760
      END IF                                                             MAI05770
C                                                                        MAI05780
      WRITE(6,309)REPS                                                   MAI05790
309   FORMAT(' NUMBER OF TIMES SIMULATION REPEATS = ',I10)               MAI05800
C                                                                        MAI05810
C     *** CALL THE STATISTICS SUBROUTINE FOR FINAL CALCULATIONS ***      MAI05820
C                                                                        MAI05830
      FLAG=1                                                             MAI05840
      CALL STATS(IN,REP,FLAG)                                            MAI05850
C                                                                        MAI05860
      IF(TYPE1.EQ.3.AND.((N-1.D0)*STEP).LE.1./BITR)THEN                  MAI05870
          WRITE(6,310)                                                   MAI05880
310       FORMAT(' LESS THAN 1 BIT WAS MODULATED DURING EACH REPETITION  MAI05890
     *    ')                                                             MAI05900
      ELSE IF(((N-1.D0)*STEP).LE.BAUDD)THEN                              MAI05910
          WRITE(6,320)                                                   MAI05920
320       FORMAT(' LESS THAN 1 BAUD WAS MODULATED DURING EACH REPETITIONMAI05930
     *    ')                                                             MAI05940
```

```
      END IF                                                            MAI05950
C                                                                       MAI05960
C     *** DETERMINE IF ANOTHER SIMULATION IS TO BE RUN ***              MAI05970
C                                                                       MAI05980
      WRITE(10,330)                                                     MAI05990
330   FORMAT(' ENTER A 1 IF YOU DESIRE TO SIMULATE ANOTHER SIGNAL.',/,  MAI06000
     *' ENTER ANY CTHER INTEGER IF YOU ARE READY TO QUIT.',//,' :')     MAI06010
C                                                                       MAI06020
      READ(5,*)ANS                                                      MAI06030
C                                                                       MAI06040
      IF(ANS.EQ.1) THEN                                                 MAI06050
C         CALL FRTCMS('CLRSCRN ')                                       MAI06060
          GO TO 29                                                      MAI06070
      END IF                                                            MAI06080
C                                                                       MAI06090
      STOP                                                              MAI06100
      END                                                               MAI06110
```

```
      SUBROUTINE BPSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,
     *DSEED,IN,REP)
C
C     *** PURPOSE ***
C
C     THIS SUBROUTINE MODULATES THE CARRIER USING BINARY PHASE SHIFT
C     KEYING AS THE MODULATION TECHNIQUE.
C
C     *** PARAMETER DEFINITIONS ***
C
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED
C     BAUD=        SYMBOL RATE OR BAUD RATE
C     FREQ=        CARRIER FREQUENCY
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES
C     AMP=         AMPLITUDE
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
C                  THE CALL THE THIS SUBROUTINE
C
C     *** VARIABLE DEFINITIONS ***
C
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
C                  AND THE FFT
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
C     TIME=        TIME VARIABLE
C     MT=          VALUE OF THE BINARY DIGIT
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
C     PI=          NUMERICAL CONSTANT
C     OMEGA=       CARRIER ANGULAR FREQUENCY
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS
C     NBAUD=       A COUNT OF THE NUMBER OF BAUDS MODULATED
C     BAUDD=       BAUD DURATION
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE
C     RMAX=        VALUE OF THE PRINCIPLE HARMONIC
C     FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
C                  SUBPROGRAM STATS
C     IND2P1=      ARRAY LENGTH DIVIDED BY 2 PLUS 1
C
C     *** VARIABLE DECLARATIONS ***
C
      INTEGER R,ANS2,TYPE2,IN,RMAX,REP,FLAG,IND2P1
C
      DOUBLE PRECISION BAUD,FREQ,IPHAS,AMP,TIME,MT,
```

```
     *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,
     *DSEED,MAXY,MINY,INT
C
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
     *SUMX4(513)
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
      COMPLEX*16 X(513)
C
      INTEGER IWK(10)
C
C     *** VARIABLE INITIALIZATION ***
C
      R=1
      TIME=0.D0
      STEP=1.D0/(2.D0*(FREQ+BAUD))
      PI=3.141592653589793D0
      OMEGA=2.D0*PI*FREQ
      DELTA=IPHAS*PI/180.D0
      NBAUD=1.D0
      BAUDD=1.D0/BAUD
      MAXY=0.D0
      IND2P1=IN/2+1
C
C     *** DO THE MODULATION AND ASSIGN THE VALUES TO ARRAY ***
C
      CALL STREAM(DSEED,ANS2,TYPE2,MT)
C
      DO 10 I=1,IN
          ARRAY(R,1)=AMP*MT*DCOS((OMEGA*TIME)+DELTA)
          ARRAY(R,2)=TIME
          IF(DABS(ARRAY(R,1)).GT.MAXY)THEN
              MAXY=DABS(ARRAY(R,1))
          END IF
          R=R+1
          TIME=TIME+STEP
          IF(TIME.GT.NBAUD*BAUDD)THEN
              CALL STREAM(DSEED,ANS2,TYPE2,MT)
              NBAUD=NBAUD+1.D0
          END IF
10    CONTINUE
C
C     *** PLOT THE TIME SERIES IF DESIRED ***
C
      IF(REP.EQ.1)THEN
          MINY=-MAXY
C
          CALL PLOT(MAXY,MINY,STEP,IN)
      END IF
C
```

```
C      *** GENERATE THE FFT ***
C
       CALL FFTRC(ARRAY(1,1),IN,X,IWK)
C
C      *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***
C      *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***
C
       N=0.D0
       R=1
       MAXY=0.D0
       DO 20 I=1,IND2P1
          ARRAY(R,1)=CDABS(X(R))
          ARRAY(R,2)=N/STEP
          IF(ARRAY(R,1).GT.MAXY)THEN
             MAXY=ARRAY(R,1)
             RMAX=R-1
          END IF
          R=R+1
          N=N+1.D0
20     CONTINUE
C
C      *** DISPLAY INFO IF THE FIRST TIME THROUGH SUBPROGRAM ***
C
       IF(REP.EQ.1)THEN
C
29     WRITE(10,30)RMAX
30     FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,
      *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,
      *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,
      *' CONTINUE WITH THE PROGRAM.',//,' :')
C
       READ(5,*)L
C
       IF(L.NE.1)THEN
          CALL FRTCMS('CLRSCRN ')
C         WRITE(10,40)
40        FORMAT(' ERROR',/)
          GO TO 29
       END IF
C
       END IF
C
C      *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***
C
       IF(REP.EQ.1)THEN
          MINY=0.D0
          INT=1.D0/STEP
C
          CALL PLOT(MAXY,MINY,INT,IND2P1)
       END IF
```

```
C                                                                                BPS01610
C     *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***          BPS01620
C                                                                                BPS01630
      FLAG=0                                                                     BPS01640
      CALL STATS(IN,REP,FLAG)                                                    BPS01650
C                                                                                BPS01660
      RETURN                                                                     BPS01670
      END                                                                        BPS01680
```

```
      SUBROUTINE DBPSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,
     *DSEED,IN,REP)
C
C     *** PURPOSE ***
C
C     THIS SUBROUTINE MODULATES THE CARRIER USING DIFFERENTIAL BINARY
C     SHIFT KEYING AS THE MODULATION TECHNIQUE.
C
C     *** PARAMETER DEFINITIONS ***
C
C     TYPE2=      INDICATES LOGIC TYPE TO BE EMPLOYED
C     BAUD=       SYMBOL RATE OR BAUD RATE
C     FREQ=       CARRIER FREQUENCY
C     IPHAS=      INITIAL PHASE ANGLE IN DEGREES
C     AMP=        AMPLITUDE
C     ANS2=       INTEGER VARIABLE TO BE PASSED TO STREAM
C     DSEED=      DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
C     IN=         NUMBER OF POSITIONS OF ARRAY TO BE UTILIZED
C                 THE CALL THE THIS SUBROUTINE
C     FLAG=       AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
C
C     *** VARIABLE DEFINITICNS ***
C
C     ARRAY=      A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
C                 DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
C                 AND THE FFT
C     R=          AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
C     TIME=       TIME VARIABLE
C     MT=         VALUE OF THE BINARY DIGIT
C     STEP=       INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
C     PI=         NUMERICAL CONSTANT
C     OMEGA=      CARRIER ANGULAR FREQUENCY
C     DELTA=      INITIAL PHASE OFFSET IN RADIANS
C     NBAUD=      A COUNT OF THE NUMBER OF BAUDS MODULATED
C     BAUDD=      BAUD DURATION
C     REF=        REFERENCE VOLTAGE
C     BK=         VALUE OF DIFFERENTIAL BIT TO BE MODULATED
C     X=          COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
C     IWK=        INTEGER ARRAY USED IN SUBROUTINE FFTRC
C     MAXY=       MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     MINY=       MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     RMAX=       PRINCIPLE HARMONIC
C     REP=        AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
C                 SUBPROGRAM STATS
C     IND2P1=     LENGTH OF ARRAY DIVIDED BY 2 + 1
C
C     *** VARIABLE DECLARATIONS ***
C
      INTEGER R,ANS2,TYPE2,IN,RMAX,REP,FLAG,IND2P1
C
```

```
      DOUBLE PRECISION BAUD,FREQ,IPHAS,AMP,TIME,MT,
     *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,
     *DSEED,REF,BK,MAXY,MINY,INT
C
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
     *SUMX4(513)
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
      COMPLEX*16 X(513)
C
      INTEGER IWK(10)
C
C     *** VARIABLE INITIALIZATION ***
C
      R=1
      TIME=0.D0
      STEP=1.D0/(2.D0*(FREQ+BAUD))
      PI=3.141592653589793D0
      OMEGA=2.D0*PI*FREQ
      DELTA=IPHAS*PI/180.D0
      NBAUD=1.D0
      BAUDD=1.D0/BAUD
      REF=1.D0
      MAXY=0.D0
      IND2P1=IN/2+1
C
C     *** DO THE MODULATION AND ASSIGN THE VALUES TO ARRAY ***
C
      CALL STREAM(DSEED,ANS2,TYPE2,MT)
C
      BK=REF*MT
C
      DO 10 I=1,IN
          ARRAY(R,1)=AMP*BK*DCOS((OMEGA*TIME)+DELTA)
          ARRAY(R,2)=TIME
          IF(ARRAY(R,1).GT.MAXY) THEN
              MAXY=ARRAY(R,1)
          END IF
          R=R+1
          TIME=TIME+STEP
          IF(TIME.GT.NBAUD*BAUDD) THEN
              CALL STREAM(DSEED,ANS2,TYPE2,MT)
              BK=BK*MT
              NBAUD=NBAUD+1.D0
          END IF
10    CONTINUE
C
C     *** ALLOW FOR THE PLOT OF THE TIME SERIES ***
C
      IF(REP.EQ.1) THEN
```

```
      MINY=-MAXY
C
         CALL PLOT(MAXY,MINY,STEP,IN)
      END IF
C
C     *** GENERATE THE FFT ***
C
      CALL FFTRC(ARRAY(1,1),IN,X,IWK)
C
C     *** PRODUCE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***
C     *** FIND THE NUMBER AND MAGNITUDE OF THE PRINCIPLE HARMONIC ***
C
      N=0.D0
      R=1
      MAXY=0.D0
      DO 20 I=1,IND2P1
         ARRAY(R,1)=CDABS(X(R))
         ARRAY(R,2)=N/STEP
         IF(ARRAY(R,1).GT.MAXY) THEN
            MAXY=ARRAY(R,1)
            RMAX=R-1
         END IF
         R=R+1
         N=N+1.D0
20    CONTINUE
C
C     *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***
C
      IF(REP.EQ.1) THEN
C
29    WRITE(10,30)RMAX
30    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,
     *' CONTINUE WITH THE PROGRAM.',//,' :')
C
      READ(5,*)L
C
      IF(L.NE.1)THEN
C        CALL FRTCMS('CLRSCRN ')
         WRITE(10,40)
40       FORMAT(' ERROR',/)
         GO TO 29
      END IF
C
      END IF
C
C     *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM ***
C
      IF(REP.EQ.1)THEN
```

```
      MINY=0.D0
      INT=1.D0/STEP
C
      CALL PLOT(MAXY,MINY,INT,IND2P1)
   END IF
C
C  *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
C
   FLAG=0
   CALL STATS(IN,REP,FLAG)
C
   RETURN
   END
```

```
      SUBROUTINE OBPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,
     *DSEED,IN,REP)
C
C     *** PURPOSE ***
C
C     THIS SUBROUTINE MODULATES THE CARRIER USING M-ARY PHASE SHIFT
C     KEYING AS THE MODULATION TECHNIQUE.
C
C     *** PARAMETER DEFINITIONS ***
C
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED
C     BAUD=        SYMBOL RATE OR BAUD RATE
C     BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD
C     FREQ=        CARRIER FREQUENCY
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES
C     AMP=         AMPLITUDE
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
C                  THE CALL THE THIS SUBROUTINE
C
C     *** VARIABLE DEFINITIONS ***
C
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
C                  AND THE FFT
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
C     TIME=        TIME VARIABLE
C     MT=          VALUE OF THE BINARY DIGIT
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
C     PI=          NUMERICAL CONSTANT
C     OMEGA=       CARRIER ANGULAR FREQUENCY
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS
C     NBIT=        A COUNT OF THE NUMBER OF BITS MODULATED
C     BITR=        BIT RATE
C     BITD=        BIT DURATION
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE
C     RMAX=        VALUE OF THE PRINCIPLE HARMONIC
C     K=           INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD
C     KM1=         INTEGER VALUE OF K MINUS 1
C     N=           NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K
C     NDP=         DOUBLE PRECISION OF N
C     ND2=         NDP DIVIDED BY 2
C     SUM1=        VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A
C                  BINARY CODE WORD OF LENGTH K
```

```
C       SUM2=          VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1
C       FLAG=          AN INTEGER WHICH CCNTROL ENTRANCE AND EXIT FROM
C                      SUBPROGRAM STATS AND ORTHO
C       IND2P1=        LENGTH OF ARRAY DIVIDED BY 2 + 1
C
C       *** VARIABLE DECLARATIONS ***
C
        INTEGER R,ANS2,TYPE2,IN,RMAX,K,KM1,REP,FLAG,IND2P1,N,C
C
        DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT,
       *STEP,PI,OMEGA,DELTA,NBIT,BITD,BITR,
       *DSEED,MAXY,MINY,INT,NDP,ND2,SUM1,SUM2
C
        DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
       *SUMX4(513)
        COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
        COMPLEX*16 X(513)
C
        INTEGER IWK(10)
C
C       *** VARIABLE INITIALIZATION ***
C
        R=1
        K=BITS
        N=2**K
        NDP=N
        NBIT=1.D0
        BITR=N*BAUD
        BITD=1.D0/BITR
        TIME=0.D0
        STEP=1.D0/(2.D0*(FREQ+BITR))
        PI=3.141592653589793D0
        OMEGA=2.D0*PI*FREQ
        DELTA=IPHAS*PI/180.D0
        MAXY=0.D0
        IND2P1=IN/2+1
C
C       *** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY
C       VARIABLE IN DECIMAL FORM ***
C
        SUM1=0.D0
        KM1=K-1
        DO 10 I=1,K
            CALL STREAM(DSEED,ANS2,TYPE2,MT)
            IF(TYPE2.EQ.1.AND.MT.EQ.-1.D0)THEN
                MT=0.D0
            ELSE IF (TYPE2.EQ.3.AND.MT.EQ.-1.D0)THEN
                MT=1.D0
            END IF
```

```
            SUM1=SUM1+(MT*(10.D0**KM1))
            KM1=KM1-1
10      CONTINUE
C
C       *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL
C       EQUIVALENT ***
C
        SUM2=0.D0
        KM1=K-1
        ND2=NDP/2.D0
        DO 20 I=1,K
            IF(SUM1/10.D0**KM1.GE.1.D0) THEN
                SUM2=SUM2+ND2
                SUM1=SUM1-10.D0**KM1
                ND2=ND2/2.D0
                KM1=KM1-1
            ELSE
                ND2=ND2/2.D0
                KM1=KM1-1
            END IF
20      CONTINUE
C
C       *** DO THE MODULATION AND ASSIGN TO ARRAY ***
C
        C=1
        FLAG=0
        CALL ORTHO(K,MT,SUM2,C,FLAG)
        FLAG=2
        DO 30 I=1,IN
            ARRAY(R,1)=AMP*MT*DCOS((OMEGA*TIME)+DELTA)
            ARRAY(R,2)=TIME
            IF(ARRAY(R,1).GT.MAXY) THEN
                MAXY=ARRAY(R,1)
            END IF
            R=R+1
            TIME=TIME+STEP
            IF(TIME.GT.NBIT*BITD) THEN
                NBIT=NBIT+1.D0
                IF(C.GT.N) THEN
C
C       *** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY
C       VARIABLE IN DECIMAL FORM ***
C
                    SUM1=0.D0
                    KM1=K-1
                    DO 40 J=1,K
                        CALL STREAM(DSEED,ANS2,TYPE2,MT)
                        IF (TYPE2.EQ.1.AND.MT.EQ.-1.D0) THEN
                            MT=0.D0
                        ELSE IF(TYPE2.EQ.3.AND.MT.EQ.-1.D0)THEN
```

```
                    MT=1.DO
                END IF
                SUM1=SUM1+(MT*(10.DO**KM1))
                KM1=KM1-1
40          CONTINUE
C
C     *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL
C     EQUIVALENT ***
C
            SUM2=0.DO
            KM1=K-1
            ND2=NDP/2.DO
            DO 50 J=1,K
                IF(SUM1/10.DO**KM1.GE.1.DO) THEN
                    SUM2=SUM2+ND2
                    SUM1=SUM1-10.DO**KM1
                    ND2=ND2/2.DO
                    KM1=KM1-1
                ELSE
                    ND2=ND2/2.DO
                    KM1=KM1-1
                END IF
50          CONTINUE
            C=1
            FLAG=1
        END IF
        CALL ORTHO(K,MT,SUM2,C,FLAG)
        FLAG=2
    END IF
30  CONTINUE
C
C     *** PLOT THE TIME SERIES IF DESIRED ***
C
      IF(REP.EQ.1)THEN
          MINY=-MAXY
C
          CALL PLOT(MAXY,MINY,STEP,IN)
      END IF
C
C     *** GENERATE THE FFT ***
C
      CALL FFTRC(ARRAY(1,1),IN,X,IWK)
C
C     *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***
C     *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***
C
      N=0.DO
      R=1
      MAXY=0.DO
      DO 60 I=1,IND2P1
```

```
      ARRAY(R,1)=CDABS(X(R))
      ARRAY(R,2)=N/STEP
      IF(ARRAY(R,1).GT.MAXY) THEN
           MAXY=ARRAY(R,1)
           RMAX=R-1
      END IF
      R=R+1
      N=N+1.D0
60    CONTINUE
C
C     *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***
C
      IF(REP.EQ.1)THEN
C
69    WRITE(10,70) RMAX
70    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,
     *' CONTINUE WITH THE PROGRAM.',//,' :')
C
      READ(5,*)L
C
      IF(L.NE.1)THEN
          CALL FRTCMS('CLRSCRN ')
C         WRITE(10,80)
80        FORMAT(' ERROR',/)
          GO TO 69
      END IF
C
      END IF
C
C     *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***
C
      IF(REP.EQ.1)THEN
          MINY=0.D0
          INT=1.D0/STEP
C
          CALL PLOT(MAXY,MINY,INT,IND2P1)
      END IF
C
C     *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
C
      FLAG=0
      CALL STATS(IN,REP,FLAG)
C
      RETURN
      END
```

```
      SUBROUTINE QPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,
     *ANS2,DSEED,IN,REP)
C
C     *** PURPOSE ***
C
C     THIS SUBROUTINE MODULATES THE CARRIER USING QUADRATURE PHASE SHIFT
C     KEYING AS THE MODULATION TECHNIQUE.
C
C     *** PARAMETER DEFINITIONS ***
C
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED
C     BAUD=        SYMBOL RATE OR BAUD RATE
C     BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD
C     FREQ=        CARRIER FREQUENCY
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES
C     AMP=         AMPLITUDE
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
C                  THE CALL THE THIS SUBROUTINE
C
C     *** VARIABLE DEFINITIONS ***
C
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
C                  AND THE FFT
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
C     TIME=        TIME VARIABLE
C     MT=          VALUE OF BINARY DIGIT PASSED FROM SUBPROGRAM STREAM
C     MT1=         VALUE OF IN-PHASE BINARY DIGIT
C     MT2=         VALUE OF QUADRATURE BINARY DIGIT
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
C     PI=          NUMERICAL CONSTANT
C     OMEGA=       CARRIER ANGULAR FREQUENCY
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS
C     NBAUD=       A COUNT OF THE NUMBER OF BAUDS MODULATED
C     BAUDD=       BAUD DURATION
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE
C     RMAX=        VALUE OF THE PRINCIPLE HARMONIC
C     FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
C                  SUBPROGRAM STATS
C     IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1
C
C     *** VARIABLE DECLARATIONS ***
C
```

```
      INTEGER R,ANS2,TYPE2,IN,RMAX,REP,FLAG,IND2P1
C
      DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT1,MT2,
     *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,
     *DSEED,MAXY,MINY,INT,MT
C
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
     *SUMX4(513)
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
      COMPLEX*16 X(513)
C
      INTEGER IWK(10)
C
C     *** VARIABLE INITIALIZATION ***
C
      R=1
      TIME=0.D0
      STEP=1.D0/(2.D0*(FREQ+BAUD))
      PI=3.141592653589793D0
      OMEGA=2.D0*PI*FREQ
      DELTA=IPHAS*PI/180.D0
      NBAUD=1.D0
      BAUDD=1.D0/BAUD
      MAXY=0.D0
      IND2P1=IN/2+1
C
C     *** DO THE MCDULATION AND ASSIGN THE VALUES TO ARRAY***
C
      CALL STREAM(DSEED,ANS2,TYPE2,MT)
      MT1=MT
C
      CALL STREAM(DSEED,ANS2,TYPE2,MT)
      MT2=MT
C
      DO 10 I=1,IN
         ARRAY(R,1)=AMP*MT1*DCOS((OMEGA*TIME)+DELTA)+
     *              AMP*MT2*DSIN((OMEGA*TIME)+DELTA)
         ARRAY(R,2)=TIME
         IF(ARRAY(R,1).GT.MAXY) THEN
            MAXY=ARRAY(R,1)
         END IF
         R=R+1
         TIME=TIME+STEP
         IF(TIME.GT.NBAUD*BAUDD) THEN
            CALL STREAM(DSEED,ANS2,TYPE2,MT)
            MT1=MT
            CALL STREAM(DSEED,ANS2,TYPE2,MT)
            MT2=MT
            NBAUD=NBAUD+1.D0
```

```
      END IF
10    CONTINUE
C
C     *** PLOT THE TIME SERIES IF DESIRED ***
C
      IF(REP.EQ.1)THEN
          MINY=-MAXY
C
          CALL PLOT(MAXY,MINY,STEP,IN)
      END IF
C
C     *** GENERATE THE FFT ***
C
      CALL FFTRC(ARRAY(1,1),IN,X,IWK)
C
C     *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***
C     *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***
C
      N=0.D0
      R=1
      MAXY=0.D0
      DO 20 I=1,IND2P1
          ARRAY(R,1)=CDABS(X(R))
          ARRAY(R,2)=N/STEP
          IF(ARRAY(R,1).GT.MAXY) THEN
              MAXY=ARRAY(R,1)
              RMAX=R-1
          END IF
          R=R+1
          N=N+1.D0
20    CONTINUE
C
C     *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***
C
      IF(REP.EQ.1) THEN
C
29    WRITE(10,30)RMAX
30    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,
     *' CONTINUE WITH THE PROGRAM.',//,' :')
C
      READ(5,*)L
C
      IF(L.NE.1)THEN
          CALL FRTCMS('CLRSCRN ')
C
          WRITE(10,40)
40        FORMAT(' ERROR',/)
          GO TO 29
      END IF
```

```
C
      END IF
C
C     *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***
C
      IF(REP.EQ.1)THEN
          MINY=0.D0
          INT=1.D0/STEP
C
          CALL PLOT(MAXY,MINY,INT,IND2P1)
      END IF
C
C     *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
C
      FLAG=0
      CALL STATS(IN,REP,FLAG)
C
      RETURN
      END
```

```
      SUBROUTINE OQPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,
     *ANS2,DSEED,IN,REP)
C
C     *** PURPOSE ***
C
C     THIS SUBROUTINE MODULATES THE CARRIER USING OFFSET QUADRATURE
C     PHASE SHIFT KEYING AS THE MODULATION TECHNIQUE.
C
C     *** PARAMETER DEFINITIONS ***
C
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED
C     BAUD=        SYMBOL RATE OR BAUD RATE
C     BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD
C     FREQ=        CARRIER FREQUENCY
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES
C     AMP=         AMPLITUDE
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
C                  THE CALL THE THIS SUBROUTINE
C
C     *** VARIABLE DEFINITIONS ***
C
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
C                  AND THE FFT
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
C     TIME=        TIME VARIABLE
C     MT=          VALUE OF BINARY DIGIT PASSED FROM SUBPROGRAM STREAM
C     MT1=         VALUE OF IN-PHASE BINARY DIGIT
C     MT2=         VALUE OF QUADRATURE BINARY DIGIT
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
C     PI=          NUMERICAL CONSTANT
C     OMEGA=       CARRIER ANGULAR FREQUENCY
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS
C     NBAUD1=      A COUNT OF THE NUMBER OF BAUDS MODULATED IN-PHASE
C     NBAUD2=      A COUNT OF THE NUMBER OF BAUDS MODULATED QUADRATURE
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE
C     RMAX=        VALUE OF THE PRINCIPLE HARMONIC
C     FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
C                  SUBPROGRAM STATS
C     IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1
C
C     *** VARIABLE DECLARATIONS ***
C
```

```
      INTEGER R,ANS2,TYPE2,IN,RMAX,REP,FLAG,IND2P1
C
      DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT1,MT2,
     *STEP,PI,OMEGA,DELTA,BAUDD,NBAUD1,NBAUD2,
     *DSEED,MAXY,MINY,INT,MT
C
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
     *SUMX4(513)
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
      COMPLEX*16 X(513)
C
      INTEGER IWK(10)
C
C     *** VARIABLE INITIALIZATION ***
C
      R=1
      STEP=1.D0/(2.D0*(FREQ+BAUD))
      PI=3.141592653589793D0
      OMEGA=2.D0*PI*FREQ
      DELTA=IPHAS*PI/180.D0
      NBAUD1=1.D0
      NBAUD2=1.D0
      BAUDD=1.D0/BAUD
      TIME=.5D0*BAUDD
      MAXY=0.D0
      IND2P1=IN/2+1
C
C     *** DO THE MODULATION AND ASSIGN THE VALUES TO ARRAY ***
C
      CALL STREAM(DSEED,ANS2,TYPE2,MT)
      MT1=MT
C
      CALL STREAM(DSEED,ANS2,TYPE2,MT)
      MT2=MT
C
      DO 10 I=1,IN
         ARRAY(R,1)=AMP*MT1*DCOS((OMEGA*TIME)+DELTA)+
     *              AMP*MT2*DSIN((OMEGA*TIME)+DELTA)
         ARRAY(R,2)=TIME
         IF(ARRAY(R,1).GT.MAXY) THEN
            MAXY=ARRAY(R,1)
         END IF
         R=R+1
         TIME=TIME+STEP
         IF(TIME.GT.NBAUD1*BAUDD) THEN
            CALL STREAM(DSEED,ANS2,TYPE2,MT)
            MT1=MT
            NBAUD1=NBAUD1+1.D0
         END IF
```

```
C
          IF(TIME.GT.((NBAUD2*BAUDD)+(BAUDD/2.D0)))THEN
               CALL STREAM(DSEED,ANS2,TYPE2,MT)
               MT2=MT
               NBAUD2=NBAUD2+1.D0
          END IF
10    CONTINUE
C
C     *** PLOT THE TIME SERIES IF DESIRED ***
C
      IF(REP.EQ.1)THEN
          MINY=-MAXY
C
          CALL PLOT(MAXY,MINY,STEP,IN)
      END IF
C
C     *** GENERATE THE FFT ***
C
      CALL FFTRC(ARRAY(1,1),IN,X,IWK)
C
C     *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***
C     *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***
C
      N=0.D0
      R=1
      MAXY=0.D0
      DO 20 I=1,IND2P1
          ARRAY(R,1)=CDABS(X(R))
          ARRAY(R,2)=N/STEP
          IF(ARRAY(R,1).GT.MAXY)THEN
               MAXY=ARRAY(R,1)
               RMAX=R-1
          END IF
          R=R+1
          N=N+1.D0
20    CONTINUE
C
C     *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***
C
      IF(REP.EQ.1)THEN
C
29    WRITE(10,30)RMAX
30    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,
     *' CONTINUE WITH THE PROGRAM.',//,' :')
C
      READ(5,*)L
C
      IF(L.NE.1)THEN
```

```
C           CALL FRTCMS('CLRSCRN ')
            WRITE(10,40)
40          FORMAT(' ERROR',/)
            GO TO 29
         END IF
C
         END IF
C
C        *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***
C
         IF(REP.EQ.1)THEN
            MINY=0.D0
            INT=1.D0/STEP
C
            CALL PLOT(MAXY,MINY,INT,IND2P1)
         END IF
C
C        *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
C
         FLAG=0
         CALL STATS(IN,REP,FLAG)
C
         RETURN
         END
```

```
      SUBROUTINE MPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,
     *DSEED,IN,REP)
C
C     *** PURPOSE ***
C
C     THIS SUBROUTINE MODULATES THE CARRIER USING M-ARY PHASE SHIFT
C     KEYING AS THE MODULATION TECHNIQUE.
C
C     *** PARAMETER DEFINITIONS ***
C
C     TYPE2=        INDICATES LOGIC TYPE TO BE EMPLOYED
C     BAUD=         SYMBOL RATE OR BAUD RATE
C     BITS=         NUMBER OF BITS IN EACH BINARY CODE WORD
C     FREQ=         CARRIER FREQUENCY
C     IPHAS=        INITIAL PHASE ANGLE IN DEGREES
C     AMP=          AMPLITUDE
C     ANS2=         INTEGER VARIABLE TO BE PASSED TO STREAM
C     DSEED=        DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
C     IN=           NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
C     REP=          AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
C                   THE CALL THE THIS SUBROUTINE
C
C     *** VARIABLE DEFINITIONS ***
C
C     ARRAY=        A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
C                   DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
C                   AND THE FFT
C     R=            AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
C     TIME=         TIME VARIABLE
C     MT=           VALUE OF THE BINARY DIGIT
C     STEP=         INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
C     PI=           NUMERICAL CONSTANT
C     OMEGA=        CARRIER ANGULAR FREQUENCY
C     DELTA=        INITIAL PHASE OFFSET IN RADIANS
C     NBAUD=        A COUNT OF THE NUMBER OF BAUDS MODULATED
C     BAUDD=        BAUD DURATION
C     X=            COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
C     IWK=          INTEGER WORKING ARRAY USED BY FUNCTION FFTRC
C     MAXY=         MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     MINY=         MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     INT=          INTERVAL BETWEEN POINTS ON THE ORDINATE
C     KMAX=         VALUE OF THE PRINCIPLE HARMONIC
C     K=            INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD
C     KM1=          INTEGER VALUE OF K MINUS 1
C     N=            NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K
C     ND2=          N DIVIDED BY 2
C     SUM1=         VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A
C                   BINARY CODE WORD OF LENGTH K
C     SUM2=         VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1
C     J=            INTEGER USED TO INDEX A LOOP
```

```
C       FLAG=        AN INTEGER WHICH CCNTROL ENTRANCE AND EXIT FROM
C                    SUBPROGRAM STATS
C       IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1
C
C       *** VARIABLE DECLARATIONS ***
C
        INTEGER R,ANS2,TYPE2,IN,RMAX,K,KM1,J,REP,FLAG,IND2P1
C
        DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT,
       *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,
       *DSEED,MAXY,MINY,INT,N,ND2,SUM1,SUM2
C
        DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
       *SUMX4(513)
        COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
        CCMPLEX*16 X(513)
C
        INTEGER IWK(10)
C
C       *** VARIABLE INITIALIZATION ***
C
        R=1
        TIME=0.D0
        STEP=1.D0/(2.D0*(FREQ+BAUD))
        PI=3.141592653589793D0
        OMEGA=2.D0*PI*FREQ
        DELTA=IPHAS*PI/180.D0
        NBAUD=1.D0
        BAUDD=1.D0/BAUD
        K=BITS
        N=2.D0**K
        MAXY=0.D0
        IND2P1=IN/2+1
C
9       IF(R.LE.IN) THEN
C
C       *** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY
C       VARIABLE IN DECIMAL FORM ***
C
           SUM1=0.D0
           KM1=K-1
           DO 10 I=1,K
              CALL STREAM(DSEED,ANS2,TYPE2,MT)
              IF(TYPE2.EQ.1.AND.MT.EQ.-1.D0) THEN
                 MT=0.D0
              ELSE IF(TYPE2.EQ.3.AND.MT.EQ.-1.D0)THEN
                 MT=1.D0
              END IF
              SUM1=SUM1+(MT*(10.D0**KM1))
```

```
            KM1=KM1-1
10      CONTINUE
C
C     *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL
C     EQUIVALENT ***
C
        SUM2=0.D0
        KM1=K-1
        ND2=N/2.D0
        DO 20 I=1,K
            IF(SUM1/10.D0**KM1.GE.1.D0)THEN
                SUM2=SUM2+ND2
                SUM1=SUM1-10.D0**KM1
                ND2=ND2/2.D0
                KM1=KM1-1
            ELSE
                ND2=ND2/2.D0
                KM1=KM1-1
            END IF
20      CONTINUE
C
C     *** DO THE MODULATION AND ASSIGN TO ARRAY ***
C
        J=R
        DO 30 I=J,IN
            ARRAY(R,1)=AMP*DCOS((OMEGA*TIME)+
     *                 ((SUM2*2.D0*PI)/N)+DELTA)
            ARRAY(R,2)=TIME
            IF(ARRAY(R,1).GT.MAXY)THEN
                MAXY=ARRAY(R,1)
            END IF
            R=R+1
            TIME=TIME+STEP
            IF(TIME.GT.NBAUD*BAUDD)THEN
                NBAUD=NBAUD+1.D0
                GO TO 9
            END IF
30      CONTINUE
      END IF
C
C     *** PLOT THE TIME SERIES IF DESIRED ***
C
      IF(REP.EQ.1)THEN
        MINY=-MAXY
C
        CALL PLOT(MAXY,MINY,STEP,IN)
      END IF
C
C     *** GENERATE THE FFT ***
C
```

```
      CALL FFTRC(ARRAY(1,1),IN,X,IWK)
C
C     *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***
C     *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***
C
      N=0.D0
      R=1
      MAXY=0.D0
      DO 40 I=1,IND2P1
           ARRAY(R,1)=CDABS(X(R))
           ARRAY(R,2)=N/STEP
           IF(ARRAY(R,1).GT.MAXY) THEN
                MAXY=ARRAY(R,1)
                RMAX=R-1
           END IF
           R=R+1
           N=N+1.D0
40    CONTINUE
C
C     *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***
C
      IF(REP.EQ.1)THEN
C
49    WRITE(10,50)RMAX
50    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,
     *' CONTINUE WITH THE PROGRAM.',//,' :')
C
      READ(5,*)L
C
      IF(L.NE.1)THEN
C          CALL FRTCMS('CLRSCRN ')
           WRITE(10,60)
60         FORMAT(' ERROR',/)
           GO TO 49
      END IF
C
      END IF
C
C     *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***
C
      IF(REP.EQ.1)THEN
           MINY=0.D0
           INT=1.D0/STEP
C
           CALL PLOT(MAXY,MINY,INT,IND2P1)
      END IF
C
C     *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
```

```
C
      FLAG=0
      CALL STATS(IN,REP,FLAG)
C
      RETURN
      END
```

```
      SUBROUTINE MASK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,
     *DSEED,IN,REP)
C
C     *** PURPOSE ***
C
C     THIS SUBROUTINE MODULATES THE CARRIER USING M-ARY AMPLITUDE SHIFT
C     KEYING AS THE MODULATION TECHNIQUE.
C
C     *** PARAMETER DEFINITIONS ***
C
C     TYPE2=        INDICATES LOGIC TYPE TO BE EMPLOYED
C     BAUD=         SYMBOL RATE OR BAUD RATE
C     BITS=         NUMBER OF BITS IN EACH BINARY CODE WORD
C     FREQ=         CARRIER FREQUENCY
C     IPHAS=        INITIAL PHASE ANGLE IN DEGREES
C     AMP=          AMPLITUDE
C     ANS2=         INTEGER VARIABLE TO BE PASSED TO STREAM
C     DSEED=        DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
C     IN=           NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
C                   THE CALL THE THIS SUBROUTINE
C     FLAG=         AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
C
C     *** VARIABLE DEFINITIONS ***
C
C     ARRAY=        A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
C                   DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
C                   AND THE FFT
C     R=            AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
C     TIME=         TIME VARIABLE
C     MT=           VALUE OF THE BINARY DIGIT
C     STEP=         INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
C     PI=           NUMERICAL CONSTANT
C     OMEGA=        CARRIER ANGULAR FREQUENCY
C     DELTA=        INITIAL PHASE OFFSET IN RADIANS
C     NBAUD=        A COUNT OF THE NUMBER OF BAUDS MODULATED
C     BAUDD=        BAUD DURATION
C     X=            COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
C     IWK=          INTEGER WORKING ARRAY USED BY FUNCTION FFTRC
C     MAXY=         MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     MINY=         MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     INT=          INTERVAL BETWEEN POINTS ON THE ORDINATE
C     RMAX=         VALUE OF THE PRINCIPLE HARMONIC
C     K=            INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD
C     KM1=          INTEGER VALUE OF K MINUS 1
C     N=            NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K
C     ND2=          N DIVIDED BY 2
C     SUM1=         VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A
C                   BINARY CODE WORD OF LENGTH K
C     SUM2=         VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1
C     J=            INTEGER USED TO INDEX A LOOP
```

```
C       AMP1=        PORTION OF AMP USED TO MODULATE THE CARRIER DURING
C                    ANY RESPECTIVE BAUD DURATION
C       REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
C                    SUBPROGRAM STATS
C       IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1
C
C       *** VARIABLE DECLARATIONS ***
C
        INTEGER R,ANS2,TYPE2,IN,RMAX,K,KM1,J,REP,FLAG,IND2P1
C
        DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT,
       *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,
       *DSEED,MAXY,MINY,INT,N,ND2,SUM1,SUM2,AMP1
C
        DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
       *SUMX4(513)
        COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
        COMPLEX*16 X(513)
C
        INTEGER IWK(10)
C
C       *** VARIABLE INITIALIZATION ***
C
        R=1
        TIME=0.D0
        STEP=1.D0/(2.D0*(FREQ+BAUD))
        PI=3.141592653589793D0
        OMEGA=2.D0*PI*FREQ
        DELTA=IPHAS*PI/180.D0
        NBAUD=1.D0
        BAUDD=1.D0/BAUD
        K=BITS
        N=2.D0**K
        MAXY=0.D0
        IND2P1=IN/2+1
C
C       IF(R.LE.IN) THEN
C
C       *** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY
C       VARIABLE IN DECIMAL FORM ***
C
            SUM1=0.D0
            KM1=K-1
            DO 10 I=1,K
                CALL STREAM(DSEED,ANS2,TYPE2,MT)
                IF(TYPE2.EQ.1.AND.MT.EQ.-1.D0) THEN
                    MT=0.D0
                ELSE IF(TYPE2.EQ.3.AND.MT.EQ.-1.D0) THEN
                    MT=1.D0
```

```
            END IF
            SUM1=SUM1+(MT*(10.D0**KM1))
            KM1=KM1-1
10      CONTINUE
C
C     *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL
C     EQUIVALENT ***
C
        SUM2=0.D0
        KM1=K-1
        ND2=N/2.D0
        DO 20 I=1,K
            IF(SUM1/10.D0**KM1.GE.1.D0) THEN
                SUM2=SUM2+ND2
                SUM1=SUM1-10.D0**KM1
                ND2=ND2/2.D0
                KM1=KM1-1
            ELSE
                ND2=ND2/2.D0
                KM1=KM1-1
            END IF
20      CONTINUE
C
C     *** DO THE MODULATION AND ASSIGN TO ARRAY ***
C
        SUM2=SUM2+1.D0
        AMP1=AMP/SUM2
        J=R
        DO 30 I=J,IN
            ARRAY(R,1)=AMP1*DCOS((OMEGA*TIME)+DELTA)
            ARRAY(R,2)=TIME
            IF(ARRAY(R,1).GT.MAXY) THEN
                MAXY=ARRAY(R,1)
            END IF
            R=R+1
            TIME=TIME+STEP
            IF(TIME.GT.NBAUD*BAUDD) THEN
                NBAUD=NBAUD+1.D0
                GO TO 9
            END IF
30      CONTINUE
      END IF
C
C     *** PLOT THE TIME SERIES IF DESIRED ***
C
      IF(REP.EQ.1) THEN
        MINY=-MAXY
C
        CALL PLOT(MAXY,MINY,STEP,IN)
      END IF
```

```
C                                                                        MAS01610
      *** GENERATE THE FFT ***                                          MAS01620
C                                                                        MAS01630
      CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                   MAS01640
C                                                                        MAS01650
C     *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***          MAS01660
C     *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***       MAS01670
C                                                                        MAS01680
      N=0.D0                                                             MAS01690
      R=1                                                                MAS01700
      MAXY=0.D0                                                          MAS01710
      DO 40 I=1,IND2P1                                                   MAS01720
         ARRAY(R,1)=CDABS(X(R))                                          MAS01730
         ARRAY(R,2)=N/STEP                                               MAS01740
         IF(ARRAY(R,1).GT.MAXY) THEN                                     MAS01750
            MAXY=ARRAY(R,1)                                              MAS01760
            RMAX=R-1                                                     MAS01770
         END IF                                                          MAS01780
         R=R+1                                                           MAS01790
         N=N+1.D0                                                        MAS01800
40    CONTINUE                                                           MAS01810
C                                                                        MAS01820
C     *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***     MAS01830
C                                                                        MAS01840
      IF(REP.EQ.1) THEN                                                  MAS01850
C                                                                        MAS01860
49    WRITE(10,50)RMAX                                                   MAS01870
50    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,  MAS01880
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,   MAS01890
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,   MAS01900
     *' CONTINUE WITH THE PROGRAM.',//,' :')                             MAS01910
C                                                                        MAS01920
      READ(5,*)L                                                         MAS01930
C                                                                        MAS01940
      IF(L.NE.1)THEN                                                     MAS01950
C        CALL FRTCMS('CLRSCRN ')                                         MAS01960
         WRITE(10,60)                                                    MAS01970
60       FORMAT(' ERROR',/)                                              MAS01980
         GO TO 49                                                        MAS01990
      END IF                                                             MAS02000
C                                                                        MAS02010
      END IF                                                             MAS02020
C                                                                        MAS02030
C     *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***   MAS02040
C                                                                        MAS02050
      IF(REP.EQ.1) THEN                                                  MAS02060
         MINY=0.D0                                                       MAS02070
         INT=1.D0/STEP                                                   MAS02080
C                                                                        MAS02090
         CALL PLCT(MAXY,MINY,INT,IND2P1)                                 MAS02100
```

```
      END IF
C
C     *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
C
      FLAG=0
      CALL STATS(IN,REP,FLAG)
C
      RETURN
      END
```

```
      SUBROUTINE QASK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,              QAS00110
     *DSEED,IN,REP)                                                     QAS00120
C                                                                       QAS00130
C     *** PURPOSE ***                                                   QAS00140
C                                                                       QAS00150
C     THIS SUBROUTINE MODULATES THE CARRIER USING QUADRATURE AMPLITUDE  QAS00160
C     SHIFT KEYING AS THE MODULATION TECHNIQUE.                         QAS00170
C                                                                       QAS00180
C     *** PARAMETER DEFINITIONS ***                                     QAS00190
C                                                                       QAS00200
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED                  QAS00210
C     BAUD=        SYMBOL RATE OR BAUD RATE                             QAS00220
C     BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD              QAS00230
C     FREQ=        CARRIER FREQUENCY                                    QAS00240
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES                       QAS00250
C     AMP=         AMPLITUDE                                            QAS00260
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM              QAS00270
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR    QAS00280
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED          QAS00290
C                  THE CALL THE THIS SUBROUTINE                         QAS00300
C     FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM      QAS00310
C                                                                       QAS00320
C     *** VARIABLE DEFINITIONS ***                                      QAS00330
C                                                                       QAS00340
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY      QAS00350
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION    QAS00360
C                  AND THE FFT                                          QAS00370
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY        QAS00380
C     TIME=        TIME VARIABLE                                        QAS00390
C     MT=          VALUE OF THE BINARY DIGIT                            QAS00400
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED           QAS00410
C     PI=          NUMERICAL CONSTANT                                   QAS00420
C     OMEGA=       CARRIER ANGULAR FREQUENCY                            QAS00430
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS                      QAS00440
C     NBAUD=       A COUNT OF THE NUMBER OF BAUDS MODULATED             QAS00450
C     BAUDD=       BAUD DURATION                                        QAS00460
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES  QAS00470
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC         QAS00480
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE          QAS00490
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE          QAS00500
C     IN1=         INTERVAL BETWEEN POINTS ON THE ORDINATE              QAS00510
C     RMAX=        VALUE OF THE PRINCIPLE HARMONIC                      QAS00520
C     K=           INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD QAS00530
C     KM1=         INTEGER VALUE OF K MINUS 1                           QAS00540
C     N=           NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K          QAS00550
C     ND2=         N DIVIDED BY 2                                       QAS00560
C     SUM1=        VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A    QAS00570
C                  BINARY CODE WORD OF LENGTH K                         QAS00580
C     SUM2=        VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1 QAS00590
C     J=           INTEGER USED TO INDEX A LOOP                         QAS00600
```

```
C       AMPA=          PORTION OF AMP USED TO MODULATE THE IN-PHASE
C                      COMPONENT OF THE CARRIER DURING ANY RESPECTIVE BAUD
C                      DURATION
C       AMPB=          PORTION OF AMP USED TO MODULATE THE QUADRATURE
C                      COMPONENT OF THE CARRIER DURING ANY RESPECTIVE BAUD
C                      DURATION
C       SUM2A=         THE VALUE OF THE IN-PHASE CODE WORD
C       SUM2B=         THE VALUE OF THE QUADRATURE CODE WORD
C       REP=           AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
C                      SUBPROGRAM STATS
C       IND2P1=        LENGTH OF ARRAY DIVIDED BY 2 + 1
C
C       *** VARIABLE DECLARATIONS ***
C
        INTEGER R,ANS2,TYPE2,IN,RMAX,K,KM1,J,M,REP,FLAG,IND2P1
C
        DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT,
       *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,
       *DSEED,MAXY,MINY,INT,N,ND2,SUM1,SUM2,AMPA,AMPB,SUM2A,SUM2B
C
        DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
       *SUMX4(513)
        COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
        COMPLEX*16 X(513)
C
        INTEGER IWK(10)
C
C       *** VARIABLE INITIALIZATION ***
C
        R=1
        TIME=0.D0
        STEP=1.D0/(2.D0*(FREQ+BAUD))
        PI=3.14159265358979 3D0
        OMEGA=2.D0*PI*FREQ
        DELTA=IPHAS*PI/180.D0
        NBAUD=1.D0
        BAUDD=1.D0/BAUD
        K=BITS
        N=2.D0**K
        MAXY=0.D0
        IND2P1=IN/2+1
C
9       IF(R.LE.IN) THEN
C
C       *** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY
C       VARIABLE IN DECIMAL FORM-DO THIS TWICE, ONCE FOR THE IN-PHASE
C       CODE WORD AND ONCE FOR THE QUADRATURE CODE WORD ***
C
            DO 16 M=1,2
```

```
C                                                                       QAS01110
            SUM1=0.D0                                                   QAS01120
            KM1=K-1                                                     QAS01130
            DO 10 I=1,K                                                 QAS01140
               CALL STREAM(DSEED,ANS2,TYPE2,MT)                         QAS01150
               IF(TYPE2.EQ.1.AND.MT.EQ.-1.D0) THEN                      QAS01160
                  MT=0.D0                                               QAS01170
               ELSE IF(TYPE2.EQ.3.AND.MT.EQ.-1.D0)THEN                  QAS01180
                  MT=1.D0                                               QAS01190
               END IF                                                   QAS01200
               SUM1=SUM1+(MT*(10.D0**KM1))                              QAS01210
               KM1=KM1-1                                                QAS01220
10          CONTINUE                                                    QAS01230
C                                                                       QAS01240
C     *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL     QAS01250
C     EQUIVALENT ***                                                    QAS01260
C                                                                       QAS01270
            SUM2=0.D0                                                   QAS01280
            KM1=K-1                                                     QAS01290
            ND2=N/2.D0                                                  QAS01300
            DO 20 I=1,K                                                 QAS01310
               IF(SUM1/10.D0**KM1.GE.1.D0) THEN                         QAS01320
                  SUM2=SUM2+ND2                                         QAS01330
                  SUM1=SUM1-10.D0**KM1                                  QAS01340
                  ND2=ND2/2.D0                                          QAS01350
                  KM1=KM1-1                                             QAS01360
               ELSE                                                     QAS01370
                  ND2=ND2/2.D0                                          QAS01380
                  KM1=KM1-1                                             QAS01390
               END IF                                                   QAS01400
20          CONTINUE                                                    QAS01410
C                                                                       QAS01420
            IF(M.EQ.1) THEN                                             QAS01430
               SUM2A=SUM2+1.D0                                          QAS01440
            ELSE IF(M.EQ.2) THEN                                        QAS01450
               SUM2B=SUM2+1.D0                                          QAS01460
            END IF                                                      QAS01470
C                                                                       QAS01480
16       CONTINUE                                                       QAS01490
C                                                                       QAS01500
C     *** DO THE MODULATION AND ASSIGN TO ARRAY ***                     QAS01510
C                                                                       QAS01520
         AMPA=AMP/SUM2A                                                 QAS01530
         AMPB=AMP/SUM2B                                                 QAS01540
         J=R                                                            QAS01550
         DO 30 I=J,IN                                                   QAS01560
            ARRAY(R,1)=AMPA*DCOS((OMEGA*TIME)+DELTA)+                   QAS01570
     *                 AMPB*DSIN((OMEGA*TIME)+DELTA)                    QAS01580
            ARRAY(R,2)=TIME                                             QAS01590
            IF(ARRAY(R,1).GT.MAXY) THEN                                 QAS01600
```

```
                MAXY=ARRAY(R,1)
             END  IF
             R=R+1
             TIME=TIME+STEP
             IF(TIME.GT.NBAUD*BAUDD) THEN
                NBAUD=NBAUD+1.DO
                GO TO 9
             END  IF
30       CONTINUE
      END IF
C
C     *** PLOT THE TIME SERIES IF DESIRED ***
C
      IF(REP.EQ.1) THEN
         MINY=-MAXY
C
         CALL PLOT(MAXY,MINY,STEP,IN)
      END IF
C
C     *** GENERATE THE FFT ***
C
      CALL FFTRC(ARRAY(1,1),IN,X,IWK)
C
C     *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***
C     *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***
C
      N=0.DO
      R=1
      MAXY=0.DO
      DO 40  I=1,IND2P1
         ARRAY(R,1)=CDABS(X(R))
         ARRAY(R,2)=N/STEP
         IF(ARRAY(R,1).GT.MAXY) THEN
            MAXY=ARRAY(R,1)
            RMAX=R-1
         END IF
         R=R+1
         N=N+1.DO
40    CONTINUE
C
C     *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***
C
      IF(REP.EQ.1) THEN
C
49    WRITE(10,50) RMAX
50    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,
     *' CONTINUE WITH THE PROGRAM.',//,' :')
C
```

```
      READ(5,*)L
C
      IF(L.NE.1)THEN
         CALL FRTCMS('CLRSCRN ')
C
         WRITE(10,60)
60       FORMAT(' ERROR',/)
         GO TO 49
      END IF
C
      END IF
C
C     *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***
C
      IF(REP.EQ.1)THEN
         MINY=0.D0
         INT=1.D0/STEP
C
         CALL PLOT(MAXY,MINY,INT,IND2P1)
      END IF
C
C     *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
C
      FLAG=0
      CALL STATS(IN,REP,FLAG)
C
      RETURN
      END
```

```
      SUBROUTINE MSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,
     *DSEED,IN,REP)
C
C     *** PURPOSE ***
C
C     THIS SUBROUTINE MODULATES THE CARRIER USING MINIMUM SHIFT KEYING
C     OR FAST FREQUENCY SHIFT KEYING AS THE MODULATION TECHNIQUE.
C
C     *** PARAMETER DEFINITIONS ***
C
C     TYPE2=      INDICATES LOGIC TYPE TO BE EMPLOYED
C     BAUD=       SYMBOL RATE OR BAUD RATE
C     FREQ=       CARRIER FREQUENCY
C     IPHAS=      INITIAL PHASE ANGLE IN DEGREES
C     AMP=        AMPLITUDE
C     ANS2=       INTEGER VARIABLE TO BE PASSED TO STREAM
C     DSEED=      DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
C     IN=         NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
C     REP=        AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
C                 THE CALL THE THIS SUBROUTINE
C
C     *** VARIABLE DEFINITIONS ***
C
C     ARRAY=      A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
C                 DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
C                 AND THE FFT
C     R=          AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
C     TIME=       TIME VARIABLE
C     MT=         VALUE OF THE BINARY DIGIT
C     STEP=       INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
C     PI=         NUMERICAL CONSTANT
C     OMEGA=      CARRIER ANGULAR FREQUENCY
C     DELTA=      INITIAL PHASE OFFSET IN RADIANS
C     NBAUD=      A COUNT OF THE NUMBER OF BAUDS MODULATED
C     BAUDD=      BAUD DURATION
C     X=          COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
C     IWK=        INTEGER WORKING ARRAY USED BY FUNCTION FFTRC
C     MAXY=       MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     MINY=       MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     INT=        INTERVAL BETWEEN POINTS ON THE ORDINATE
C     RMAX=       VALUE OF THE PRINCIPLE HARMONIC
C     FLAG=       AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
C                 SUBPROGRAM STATS
C     IND2P1=     LENGTH OF ARRAY DIVIDED BY 2 + 1
C
C     *** VARIABLE DECLARATIONS ***
C
      INTEGER R,ANS2,TYPE2,IN,RMAX,REP,FLAG,IND2P1
C
      DOUBLE PRECISION BAUD,FREQ,IPHAS,AMP,TIME,MT,
```

```
      *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,
      *DSEED,MAXY,MINY,INT
C
       DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
      *SUMX4(513)
       COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
       COMPLEX*16 X(513)
C
       INTEGER IWK(10)
C
C      *** VARIABLE INITIALIZATION ***
C
       R=1
       TIME=0.D0
       STEP=1.D0/(2.D0*(FREQ+(1.25D0*BAUD)))
       PI=3.141592653589793D0
       OMEGA=2.D0*PI*FREQ
       DELTA=IPHAS*PI/180.D0
       NBAUD=1.D0
       BAUDD=1.D0/BAUD
       MAXY=0.D0
       IND2P1=IN/2+1
C
       CALL STREAM(DSEED,ANS2,TYPE2,MT)
C
       DO 10 I=1,IN
          ARRAY(R,1)=AMP*DCOS(((OMEGA+((MT*PI)/(2.D0*BAUDD)))*TIME)
      *   +DELTA)
          ARRAY(R,2)=TIME
          IF(DABS(ARRAY(R,1)).GT.MAXY) THEN
             MAXY=DABS(ARRAY(R,1))
          END IF
          R=R+1
          TIME=TIME+STEP
          IF(TIME.GT.NBAUD*BAUDD) THEN
             CALL STREAM(DSEED,ANS2,TYPE2,MT)
             NBAUD=NBAUD+1.D0
          END IF
10     CONTINUE
C
C      *** PLOT THE TIME SERIES IF DESIRED ***
C
       IF(REP.EQ.1) THEN
          MINY=-MAXY
C
          CALL PLOT(MAXY,MINY,STEP,IN)
       END IF
C
C      *** GENERATE THE FFT ***
```

```
C
      CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                   MSK01110
C                                                                       MSK01120
C     *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***          MSK01130
C     *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***       MSK01140
C                                                                       MSK01150
      N=0.D0                                                            MSK01160
      R=1                                                               MSK01170
      MAXY=0.D0                                                         MSK01180
      DO 20 I=1,IND2P1                                                  MSK01190
          ARRAY(R,1)=CDABS(X(R))                                        MSK01200
          ARRAY(R,2)=N/STEP                                             MSK01210
          IF(ARRAY(R,1).GT.MAXY) THEN                                   MSK01220
              MAXY=ARRAY(R,1)                                           MSK01230
              RMAX=R-1                                                  MSK01240
          END IF                                                        MSK01250
          R=R+1                                                         MSK01260
          N=N+1.D0                                                      MSK01270
20    CONTINUE                                                          MSK01280
C                                                                       MSK01290
C     *** DISPLAY INFO IF THE FIRST TIME THROUGH SUBPROGRAM ***         MSK01300
C                                                                       MSK01310
      IF(REP.EQ.1)THEN                                                  MSK01320
C                                                                       MSK01330
29    WRITE(10,30)RMAX                                                  MSK01340
30    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,  MSK01350
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,   MSK01360
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,   MSK01370
     *' CONTINUE WITH THE PROGRAM.',//,' :')                            MSK01380
C                                                                       MSK01390
      READ(5,*)L                                                        MSK01400
C                                                                       MSK01410
      IF(L.NE.1)THEN                                                    MSK01420
          CALL FRTCMS('CLRSCRN ')                                       MSK01430
C                                                                       
          WRITE(10,40)                                                  MSK01440
40        FORMAT(' ERROR',/)                                            MSK01450
          GO TO 29                                                      MSK01460
      END IF                                                            MSK01470
C                                                                       MSK01480
      END IF                                                            MSK01490
C                                                                       MSK01500
C     *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***   MSK01510
C                                                                       MSK01520
      IF(REP.EQ.1)THEN                                                  MSK01530
          MINY=0.D0                                                     MSK01540
          INT=1.D0/STEP                                                 MSK01550
C                                                                       MSK01560
          CALL PLOT(MAXY,MINY,INT,IND2P1)                               MSK01570
      END IF                                                            MSK01580
C                                                                       MSK01590
                                                                        MSK01600
```

```
C     *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***   MSK01610
C                                                                       MSK01620
      FLAG=0                                                            MSK01630
      CALL STATS(IN,REP,FLAG)                                           MSK01640
C                                                                       MSK01650
      RETURN                                                            MSK01660
      END                                                               MSK01670
```

```
      SUBROUTINE MFSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,
     *DSEED,IN,REP)
C
C     *** PURPOSE ***
C
C     THIS SUBROUTINE MODULATES THE CARRIER USING M-ARY FREQUENCY SHIFT
C     KEYING AS THE MODULATION TECHNIQUE.
C
C     *** PARAMETER DEFINITIONS ***
C
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED
C     BAUD=        SYMBOL RATE OR BAUD RATE
C     BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD
C     FREQ=        CARRIER FREQUENCY
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES
C     AMP=         AMPLITUDE
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
C                  THE CALL THE THIS SUBROUTINE
C
C     *** VARIABLE DEFINITIONS ***
C
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
C                  AND THE FFT
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
C     TIME=        TIME VARIABLE
C     MT=          VALUE OF THE BINARY DIGIT
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
C     PI=          NUMERICAL CONSTANT
C     OMEGA=       CARRIER ANGULAR FREQUENCY
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS
C     NBAUD=       A COUNT OF THE NUMBER OF BAUDS MODULATED
C     BAUDD=       BAUD DURATION
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE
C     RMAX=        VALUE OF THE PRINCIPLE HARMONIC
C     K=           INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD
C     KM1=         INTEGER VALUE OF K MINUS 1
C     N=           NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K
C     ND2=         N DIVIDED BY 2
C     SUM1=        VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A
C                  BINARY CODE WORD OF LENGTH K
C     SUM2=        VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1
C     J=           INTEGER USED TO INDEX A LOOP
```

```
C       FLAG=         AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
C                     SUBPROGRAM STATS
C       DELF=         DELTA FREQUENCY BETWEEN SUCCESSIVE CODE WORDS
C       FRNG=         FREQUENCY RANGE AROUND THE CENTRAL CARRIER FREQUENCY
C       FRNGD2=       FREQUENCY RANGE DIVIDED BY 2.
C       MFREQ=        FREQUENCY OF THE CODE WORD TO BE MODULATED ONTO THE
C                     CARRIER
C       IND2P1=       LENGTH OF ARRAY DIVIDED BY 2 + 1
C
C       *** VARIABLE DECLARATIONS ***
C
        INTEGER R,ANS2,TYPE2,IN,RMAX,K,KM1,J,REP,FLAG,IND2P1
C
        DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT,
       *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,
       *DSEED,MAXY,MINY,INT,N,ND2,SUM1,SUM2,DELF,FRNG,FRNGD2,MFREQ
C
        DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
       *SUMX4(513)
        COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
        COMPLEX*16 X(513)
C
        INTEGER IWK(10)
C
C       *** VARIABLE INITIALIZATION ***
C
        R=1
        TIME=0.D0
        BAUDD=1.D0/BAUD
        DELF=1.D0/BAUDD
        K=BITS
        N=2.D0**K
        FRNG=(N-1.D0)*DELF
        FRNGD2=FRNG/2.D0
        STEP=1.D0/(2.D0*(FREQ+FRNGD2+BAUD))
        PI=3.141592653589793D0
        OMEGA=2.D0*PI*FREQ
        DELTA=IPHAS*PI/180.D0
        NBAUD=1.D0
        MAXY=0.D0
        IND2P1=IN/2+1
C
9       IF(R.LE.IN) THEN
C
C       *** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY
C       VARIABLE IN DECIMAL FORM ***
C
            SUM1=0.D0
            KM1=K-1
```

```
      DO 10 I=1,K
         CALL STREAM(DSEED,ANS2,TYPE2,MT)
         IF(TYPE2.EQ.1.AND.MT.EQ.-1.D0) THEN
            MT=0.D0
         ELSE IF(TYPE2.EQ.3.AND.MT.EQ.-1.D0) THEN
            MT=1.D0
         END IF
         SUM1=SUM1+(MT*(10.D0**KM1))
         KM1=KM1-1
10    CONTINUE
C
C  *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL
C  EQUIVALENT ***
C
      SUM2=0.D0
      KM1=K-1
      ND2=N/2.D0
      DO 20 I=1,K
         IF(SUM1/10.D0**KM1.GE.1.D0) THEN
            SUM2=SUM2+ND2
            SUM1=SUM1-10.D0**KM1
            ND2=ND2/2.D0
            KM1=KM1-1
         ELSE
            ND2=ND2/2.D0
            KM1=KM1-1
         END IF
20    CONTINUE
C
C  *** DO THE MODULATION AND ASSIGN TO ARRAY ***
C
      J=R
      MFREQ=-FRNGD2+(SUM2*DELF)
      DO 30 I=J,IN
         ARRAY(R,1)=AMP*DCOS(((OMEGA+(2.D0*PI*MFREQ))*TIME)+
     *               DELTA)
         ARRAY(R,2)=TIME
         IF(ARRAY(R,1).GT.MAXY) THEN
            MAXY=ARRAY(R,1)
         END IF
         R=R+1
         TIME=TIME+STEP
         IF(TIME.GT.NBAUD*BAUDD) THEN
            NBAUD=NBAUD+1.D0
            GO TO 9
         END IF
30    CONTINUE
   END IF
C
C  *** PLOT THE TIME SERIES IF DESIRED ***
```

```
C                                                                       MFS01610
      IF(REP.EQ.1)THEN                                                  MFS01620
          MINY=-MAXY                                                    MFS01630
C                                                                       MFS01640
          CALL PLOT(MAXY,MINY,STEP,IN)                                  MFS01650
      END IF                                                            MFS01660
C                                                                       MFS01670
C     *** GENERATE THE FFT ***                                          MFS01680
C                                                                       MFS01690
      CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                   MFS01700
C                                                                       MFS01710
C     *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***          MFS01720
C     *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***       MFS01730
C                                                                       MFS01740
      N=0.D0                                                            MFS01750
      R=1                                                               MFS01760
      MAXY=0.D0                                                         MFS01770
      DO 40 I=1,IND2P1                                                  MFS01780
          ARRAY(R,1)=CDABS(X(R))                                        MFS01790
          ARRAY(R,2)=N/STEP                                             MFS01800
          IF(ARRAY(R,1).GT.MAXY)THEN                                    MFS01810
              MAXY=ARRAY(R,1)                                           MFS01820
              RMAX=R-1                                                  MFS01830
          END IF                                                        MFS01840
          R=R+1                                                         MFS01850
          N=N+1.D0                                                      MFS01860
40    CONTINUE                                                          MFS01870
C                                                                       MFS01880
C     *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***     MFS01890
C                                                                       MFS01900
      IF(REP.EQ.1)THEN                                                  MFS01910
C                                                                       MFS01920
49    WRITE(10,50)RMAX                                                  MFS01930
50    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,  MFS01940
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,   MFS01950
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,   MFS01960
     *' CONTINUE WITH THE PROGRAM.',//,' :')                            MFS01970
C                                                                       MFS01980
      READ(5,*)L                                                        MFS01990
C                                                                       MFS02000
      IF(L.NE.1)THEN                                                    MFS02010
C         CALL FRTCMS('CLRSCRN ')                                       MFS02020
          WRITE(10,60)                                                  MFS02030
60        FORMAT(' ERROR',/)                                            MFS02040
          GO TO 49                                                      MFS02050
      END IF                                                            MFS02060
C                                                                       MFS02070
      END IF                                                            MFS02080
C                                                                       MFS02090
C     *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***   MFS02100
```

```
C                                                                       MFS02110
      IF(REP.EQ.1)THEN                                                  MFS02120
         MINY=0.D0                                                      MFS02130
         INT=1.D0/STEP                                                  MFS02140
C                                                                       MFS02150
         CALL PLOT(MAXY,MINY,INT,IND2P1)                                MFS02160
      END IF                                                            MFS02170
C                                                                       MFS02180
      FLAG=0                                                            MFS02190
      CALL STATS(IN,REP,FLAG)                                           MFS02200
C                                                                       MFS02210
      RETURN                                                            MFS02220
      END                                                               MFS02230
```

```
      SUBROUTINE QPRS(TYPE2,TYPE3,BAUD,BITS,FREQ,IPHAS,AMP,
     *ANS2,DSEED,IN,REP)
C
C     *** PURPOSE ***
C
C     THIS SUBROUTINE MODULATES THE CARRIER USING QUADRATURE PHASE SHIFT
C     KEYING AS THE MODULATION TECHNIQUE.
C
C     *** PARAMETER DEFINITIONS ***
C
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED
C     TYPE3=       INDICATES THE CLASS OF QPRS
C     BAUD=        SYMBOL RATE OR BAUD RATE
C     BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD
C     FREQ=        CARRIER FREQUENCY
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES
C     AMP=         AMPLITUDE
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
C                  THE CALL THE THIS SUBROUTINE
C
C     *** VARIABLE DEFINITIONS ***
C
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
C                  AND THE FFT
C     ASUBN=       A DOUBLE PRECISION ARRAY CONTAINING THE VALUES OF
C                  THE IN PHASE BINARY DIGIT TO BE MODULATED
C     BSUBN=       A DOUBLE PRECISION ARRAY CONTAINING THE VALUES OF THE
C                  QUADRATURE BINARY DIGIT TO BE MODULATED
C     HOFT=        THE IMPULSE RESPONSE FOR THE SPECIFIED CLASS FILTER
C     T=           VARIABLE WHICH DETERMINES THE MAGNITUDE OF THE IMPULSE
C                  RESPONSE AT A GIVEN TIME FOR A SPECIFIED BINARY DIGIT
C     NDP=         VARIABLE USED IN THE COMPUTATION OF T
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
C     TIME=        TIME VARIABLE
C     MT=          VALUE OF BINARY DIGIT PASSED FROM SUBPROGRAM STREAM
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
C     PI=          NUMERICAL CONSTANT
C     OMEGA=       CARRIER ANGULAR FREQUENCY
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS
C     BAUDD=       BAUD DURATION AND QPRS BIT DURATION
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE
C     RMAX=        VALUE OF THE PRINCIPLE HARMONIC
```

```
C     FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
C                  SUBPROGRAM STATS
C     IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1
C     MAX=         MAX NUMBER OF INCREMENTS IN 6 BAUD DURATIONS
C     INDEX=       INDEX OF A LOOP
C
C     *** VARIABLE DECLARATIONS ***
C
      INTEGER R,ANS2,TYPE2,TYPE3,IN,RMAX,REP,FLAG,IND2P1,MAX,INDEX
C
      DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT1,MT2,
     *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,HOFT,T,NDP,
     *DSEED,MAXY,MINY,INT,MT
C
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
     *SUMX4(513),ASUBN(2000),BSUBN(2000)
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
      COMPLEX*16 X(513)
C
      INTEGER IWK(10)
C
C     *** VARIABLE INITIALIZATION ***
C
      R=1
      PI=3.141592653589793D0
      BAUDD=1.D0/BAUD
      TIME=6.D0/BAUD
      STEP=1.D0/(2.D0*(FREQ+((PI+1.D0)*BAUD)))
      MAX=TIME/STEP+1.D0
      INDEX=IN+MAX
      OMEGA=2.D0*PI*FREQ
      DELTA=IPHAS*PI/180.D0
      MAXY=0.D0
      IND2P1=IN/2+1
C
      DO 10 I=1,INDEX
          CALL STREAM(DSEED,ANS2,TYPE2,MT)
          ASUBN(I)=MT
          CALL STREAM(DSEED,ANS2,TYPE2,MT)
          BSUBN(I)=MT
10    CONTINUE
C
      DO 20 I=1,1024
          ARRAY(I,1)=0.D0
20    CONTINUE
C
C     *** DO THE MODULATION AND ASSIGN THE VALUES TO ARRAY***
C
      DO 30 I=1,IN
```

```
C
      DO 40 J=1,INDEX
         NDP=J-1
         T=TIME-(NDP/BAUD)
         IF(TYPE3.EQ.1)THEN
            IF(T.EQ.-BAUDD/2.D0.OR.T.EQ.BAUDD/2.D0)THEN
               HOFT=1.D0
            ELSE
               HOFT=(4.D0/(BAUD**2*PI))*(DCOS(PI*T*BAUD)/
     *              ((1.D0/BAUD**2)-(4.D0*T**2)))
            END IF
         ELSE IF(TYPE3.EQ.2)THEN
            IF(T.EQ.0.D0) THEN
               HOFT=2.D0
            ELSE IF(T.EQ.-BAUDD.OR.T.EQ.BAUDD)THEN
               HOFT=1.D0
            ELSE
               HOFT=(2.D0/(BAUD**3*PI*T))*(DSIN(PI*T*BAUD)/
     *              ((1.D0/BAUD**2)-T**2))
            END IF
         ELSE IF(TYPE3.EQ.3)THEN
            IF(T.EQ.0.D0) THEN
               HOFT=1.D0
            ELSE IF(T.EQ.-BAUDD)THEN
               HOFT=2.D0
            ELSE IF(T.EQ.BAUDD)THEN
               HOFT=-1.D0
            ELSE
               HOFT=(1.D0/(BAUD**2*PI*T))*(DSIN(PI*T*BAUD))*
     *              ((3.D0*T-BAUDD)/(T**2-(1.D0/BAUD**2)))
            END IF
         ELSE IF(TYPE3.EQ.4)THEN
            IF(T.EQ.-BAUDD) THEN
               HOFT=1.D0
            ELSE IF(T.EQ.BAUDD)THEN
               HOFT=-1.D0
            ELSE
               HOFT=(2.D0/(BAUD**2*PI))*(DSIN(PI*T*BAUD)/
     *              (T**2-(1.D0/BAUD**2)))
            END IF
         ELSE IF(TYPE3.EQ.5)THEN
            IF(T.EQ.0.D0) THEN
               HOFT=-2.D0
            ELSE IF(T.EQ.-2.D0*BAUDD.OR.T.EQ.2.D0*BAUDD)THEN
               HOFT=1.D0
            ELSE
               HOFT=(8.D0/(BAUD**3*PI*T))*(DSIN(PI*T*BAUD)/
     *              (T**2-(4.D0/BAUD**2)))
            END IF
C
```

```
              END IF
C
              ARRAY(I,1)=ARRAY(I,1)+((ASUBN(J)*HOFT*AMP*DCOS((OMEGA*
     *                   TIME)+DELTA))+(BSUBN(J)*HOFT*AMP*DSIN((OMEGA*
     *                   TIME)+DELTA)))
C
              IF(DABS(ARRAY(I,1)).GT.MAXY)THEN
                 MAXY=DABS(ARRAY(I,1))
              END IF
C
40         CONTINUE
C
           ARRAY(I,2)=TIME
           TIME=TIME+STEP
C
30      CONTINUE
C
C       *** PLOT THE TIME SERIES IF DESIRED ***
C
        IF(REP.EQ.1)THEN
            MINY=-MAXY
C
            CALL PLOT(MAXY,MINY,STEP,IN)
        END IF
C
C       *** GENERATE THE FFT ***
C
        CALL FFTRC(ARRAY(1,1),IN,X,IWK)
C
C       *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***
C       *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***
C
        N=0.D0
        R=1
        MAXY=0.D0
        DO 50 I=1,IND2P1
            ARRAY(R,1)=CDABS(X(R))
            ARRAY(R,2)=N/STEP
            IF(ARRAY(R,1).GT.MAXY) THEN
                MAXY=ARRAY(R,1)
                RMAX=R-1
            END IF
            R=R+1
            N=N+1.D0
50      CONTINUE
C
C       *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***
C
        IF(REP.EQ.1)THEN
C
```

```
59    WRITE(10,60) RMAX
60    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,
     *' CONTINUE WITH THE PROGRAM.',//,' :')
C
      READ(5,*) L
C
      IF(L.NE. 1)THEN
         CALL FRTCMS('CLRSCRN ')
C        WRITE(10,70)
70       FORMAT(' ERROR',/)
         GO TO 59
      END IF
C
      END IF
C
C     *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***
C
      IF(REP.EQ.1)THEN
         MINY=0.D0
         INT=1.D0/STEP
C
         CALL PLOT(MAXY,MINY,INT,IND2P1)
      END IF
C
C     *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
C
      FLAG=0
      CALL STATS(IN,REP,FLAG)
C
      RETURN
      END
```

```
      SUBROUTINE STREAM(DSEED,ANS2,TYPE2,MT)
C
C     *** PURPOSE ***
C
C     THIS SUBPROGRAM ALLOWS THE INPUT OF SUCCESSIVE BITS DURING
C     MODULATION
C
C     *** PARAMETER DEFINITIONS ***
C
C     DSEED=        DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
C     ANS2=         WHETHER OR NOT THE BIT STREAM IS TO BE GENERATED
C                   BY THE RANDOM NUMBER GENERATOR
C     TYPE2=        THE TYPE OF BINARY LOGIC TO BE EMPLOYED
C     MT=           VALUE OF THE BINARY DIGIT TO BE PASSED
C
C     *** VARIABLE DEFINITIONS ***
C
C     GGUBFS=       RANDOM NUMBER GENERATOR
C     IMT=          INTEGER VALUE OF BINARY DIGIT TO BE MODULATED
C
C     *** VARIABLE DECLARATIONS ***
C
      INTEGER ANS2,TYPE2,IMT
C
      DOUBLE PRECISION DSEED,MT
C
      IF(ANS2.EQ.1)THEN
         IF(TYPE2.EQ.1)THEN
            IF(GGUBFS(DSEED).LE..5D0)THEN
               MT=1.D0
            ELSE
               MT=-1.D0
            END IF
         ELSE IF(TYPE2.EQ.2)THEN
            IF(GGUBFS(DSEED).LE..5D0)THEN
               MT=1.D0
            ELSE
               MT=0.D0
            END IF
         ELSE IF(TYPE2.EQ.3)THEN
            IF(GGUBFS(DSEED).LE..5D0)THEN
               MT=-1.D0
            ELSE
               MT=0.D0
            END IF
         END IF
      ELSE IF(ANS2.EQ.2)THEN
         WRITE(10,10)
10       FORMAT(' ENTER THE NEXT BIT IN THE DESIRED BIT STREAM.',/)
         READ(5,*)IMT
```

```
      MT=IMT                                                            STR00610
   END IF                                                               STR00620
   RETURN                                                               STR00630
   END                                                                  STR00640
```

```
      SUBROUTINE ORTHO(K,MT,SUM2,C,FLAG)
C
C     *** PURPOSE ***
C
C     THIS SUBROUTINE GENERATES AN ORTHOGONAL SET OF N VECTORS OF LENGTH
C     N, WHERE N IS 2**K, WHERE K IS THE LENGTH OF THE BINARY CODE WORD.
C     AND DETERMINES THE VALUE, MT, OF A BINARY DIGIT TO BE RETURNED
C
C     *** PARAMETER DEFINITIONS ***
C
C     K=           NUMBER OF BITS IN THE BINARY CODE WORD
C     MT=          THE VALUE OF THE BINARY DIGIT TO BE MODULATED
C     SUM2=        THE DECIMAL EQUIVALENT OF A SET OF RANDOMLY DRAWN
C                  BINARY DIGITS
C     C=           THE COLUMN OF THE VECTOR WHERE THE BINARY DIGIT TO
C                  BE MODULATED IS LOCATED
C     FLAG=        INTEGER WHICH CONTROLS POINT OF ENTRY TO SUBROUTINE
C                  CRTHO
C
C     *** VARIABLE DEFINITICNS ***
C
C     ROW=         RCW OF ARRAY H2N WHICH MATCHES A BINARY CODE WORD
C     KM1=         INTEGER VALUE OF K MINUS 1
C     N=           NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K
C     H2N=         MATRIX OF N ORTHOGONAL VECTORS OF LENGTH N
C     RH2N=        ROW OF MATRIX H2N
C     CH2N=        COLUMN OF MATRIX H2N
C     H2NR=        RANK OF MATRIX H2N
C     RH2NP1=      ROW OF MATRIX H2N EQUAL TO HALF THE RANK PLUS 1
C     CH2NP1=      COLUMN OF MATRIX H2N EQUAL TO HALF THE RANK PLUS 1
C     X=           DOUBLE PRECISION EQUIVALENT ROW REPRESENTATION OF H2N
C
C     *** VARIABLE DECLARATIONS ***
C
      INTEGER K,RH2N,CH2N,RH2NP1,CH2NP1,H2NR,N,KM1,H2N(65,65),
     *FLAG,ROW,C
C
      DOUBLE PRECISION X,MT,SUM2
C
C     *** VARIABLE INITIALIZATION ***
C
      KM1=K-1
      N=2**K
      RH2N=1
      CH2N=1
      H2NR=2
      RH2NP1=H2NR+1
      CH2NP1=H2NR+1
      H2N(1,1)=1
      H2N(1,2)=1
```

```
      H2N(2,1)=1
      H2N(2,2)=-1
C
C     *** GENERATE A MATRIX OF N ORTHOGONAL VECTORS OF LENGTH N ***
C
      IF(FLAG.EQ.0)THEN
      DO 10 I=1,KM1
         DO 20 L=1,H2NR
            DO 30 M=1,H2NR
               H2N(RH2N,CH2N)=H2N(RH2N,CH2N)
               H2N(RH2N,CH2NP1)=H2N(RH2N,CH2N)
               H2N(RH2NP1,CH2N)=H2N(RH2N,CH2N)
               H2N(RH2NP1,CH2NP1)=-H2N(RH2N,CH2N)
               CH2N=CH2N+1
               CH2NP1=CH2NP1+1
30          CONTINUE
            RH2N=RH2N+1
            RH2NP1=RH2NP1+1
            CH2N=1
            CH2NP1=H2NR+1
20       CONTINUE
         H2NR=2*H2NR
         RH2N=1
         CH2N=1
         RH2NP1=H2NR+1
         CH2NP1=H2NR+1
10    CONTINUE
C
C     *** ASSIGN EACH ROW OF H2N A REPRESENTATIVE DECIMAL NUMBER ***
C
      DO 40 I=1,N
         H2N(I,65)=I-1
40    CONTINUE
C
C     *** DETERMINE WHICH RCW OR H2N MATCHES THE VALUE OF THE
C     REPRESENTATIVE DECIMAL EQUIVALENT TO THE BINARY CODE WORD ***
C
      DO 50 I=1,N
         X=H2N(I,65)
         IF(SUM2.EQ.X)THEN
            ROW=I
         END IF
50    CONTINUE
C
      END IF
C
C     *** DETERMINE THE NEW ROW OF H2N ***
C
      IF(FLAG.EQ.1)THEN
         DO 60 I=1,N
```

```
            X=H2N(I,65)
            IF(SUM2.EQ.X) THEN
               ROW=I
            END IF
60       CONTINUE
      END IF
C
C     *** DETERMINE THE VALUE OF MT TO BE RETURNED ***
C
      MT=H2N(ROW,C)
      C=C+1
C
      RETURN
      END
```

```
      SUBROUTINE PLOT(MAXY,MINY,INT,IN)
C
C     *** PURPOSE ***
C
C     THIS SUBPROGRAM CONTROL THE USERS ABILITY TO MAKE A PLOT OF THE
C     TIME FUNCTION OR THE FFT
C
C     *** PARAMETER DEFINITIONS ***
C
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
C     INT=         DELTA TIME AT WHICH THE FUNCTION IS REPRODUCED
C     IN=          MAX LENGTH OF VECTOR TO BE PLOTTED
C
C     *** VARIABLE DEFINITIONS ***
C
C     ANS=         RESPONSE TO INQUIRY
C     ARRAY=       ARRAY CONTAINING VALUES TO BE PLOTTED
C     RANGE=       ARRAY TO SPECIFY MIN AND MAX VALUES FOR THE ORDINATE
C                  AND ABCISSA OF THE PLOT
C     NI=          INTEGER NUMBER OF X,Y PAIRS TO BE PLOTTED
C     ND=          DOUBLE PRECISION OF NI
C     R=           INTEGER VALUE FOR ROW OF ARRAY
C
C     *** VARIABLE DECLARATIONS ***
C
      DOUBLE PRECISION MAXY,MINY,INT,ND
C
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
     *SUMX4(513)
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
      REAL RANGE(4)
C
      INTEGER ANS,NI,R,IN
C
      CALL FRTCMS('CLRSCRN ')
C
      WRITE(10,10)
10    FORMAT(' ENTER A 1 IF YOU DESIRE TO GENERATE A PLOT.',/,
     *' ENTER ANY OTHER INTEGER TO CONTINUE THE PROGRAM.',//,' :')
C
      READ(5,*)ANS
C
      IF(ANS.EQ.1)THEN
         CALL FRTCMS('CLRSCRN ')
19       WRITE(10,20)IN,IN
20       FORMAT(' YOU MAY PLOT A MAXIMUM OF',I5,' X,Y PAIRS ON YOUR'
     *   ,/,' GRAPH.  IT IS RECOMMENDED THAT YOU PLOT NO MORE THAN'
     *   ,/,' 129 CONSECUTIVE PAIRS OF COORDINATE POINTS SO YOU',/,
```

```
     *    ' WILL HAVE THE OPPORTUNITY TO PLOT MORE THAN ONE SECTION',/,
     *    ' OF THE TOTAL RECORD OF',I5,' POINTS.  ENTER THE NUMBER',/,
     *    ' OF POINTS TO BE PLOTTED.',//,' :')
C
      READ(5,*)NI
C
      IF(NI.LE.IN)THEN
         CONTINUE
      ELSE
C        CALL FRTCMS('CLRSCRN ')
         WRITE(10,30)
30       FORMAT(' ERROR',/)
         GO TO 19
      END IF
C
C     CALL FRTCMS('CLRSCRN ')
39    WRITE(10,40)IN
40    FORMAT(' AT WHICH NUMBER OF THE',I5,' X,Y PAIRS DO YOU WISH',/
     *    ' TO START THE PLOT.',//,
     *    ' CAUTION!  THE POINT YOU SPECIFY TO START THE PLOT MUST BE',
     *    /,' SMALL ENOUGH TO ALLOW THE ENTIRE RANGE OF POINTS YOU',/,
     *    ' DESIRED TO HAVE PLOTTED AS SPECIFIED BY YOUR PREVIOUS INPUT'
     *    )
C
      READ(5,*)R
C
      IF(R.GT.((IN+1)-NI))THEN
C        CALL FRTCMS('CLRSCRN ')
         WRITE(10,50)
50       FORMAT(' ERROR',/)
         GO TO 39
      END IF
C
      ND=NI
      RANGE(1)=(ARRAY(R,2)+(ND*INT))
      RANGE(2)=ARRAY(R,2)
      RANGE(3)=1.5D0*MAXY
      RANGE(4)=1.5D0*MINY
C
      WRITE(10,60)
60    FORMAT('1')
      CALL UTPLOT(ARRAY(R,2),ARRAY(R,1),NI,RANGE,2,0)
C
      WRITE(10,70)
70    FORMAT(' IF YOU WOULD LIKE ANOTHER PLOT OF THE SAME GRAPH OR',
     *    /,' A PLOT OVER A DIFFERENT RANGE OR FROM ANOTHER STARTING',/,
     *    ' POINT, ENTER A 1.  ENTER ANY OTHER INTEGER TO CONTINUE WITH'
     *    ,/,' THE PROGRAM.',//,' :')
C
      READ(5,*)ANS
```

```
C
          IF(ANS.EQ.1)THEN
              GO TO 19
          END IF
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE STATS(IN,REP,FLAG)
C
C     *** PURPOSE ***
C
C     THIS SUBPROGRAM COMPILES THE ACCUMULATED STATISTICS OF THE
C     ELEMENTS OF THE AMPLITUDE SPECTRUM OF THE FFTS.
C
C     *** PARAMETER DEFINITIONS ***
C
C     IN=           INTEGER OF THE NUMBER OF POSITIONS USED IN ARRAY
C     REP=          INTEGER OF THE NUMBER OF THE REPETITION OF THE
C                   GENERATION OF THE AMPLITUDE SPECTRUM OF THE FFT
C     FLAG=         AN INTEGER THAT CONTROLS THE POINT OF ENTRANCE OR
C                   EXIT FROM THE SUBROUTINE
C
C     *** VARIABLE DEFINITIONS ***
C
C     IND2P1=       POSITIONS USED IN ARRAYS SUMX AND SUMXSQ
C     ARRAY=        AN ARRAY CONTAINING THE VALUES OF THE AMPLITUDE
C                   SPECTRUM OF THE FFT
C     SUMX=         AN ARRAY CONTAINING THE SUM OF THE VALUES OF THE
C                   AMPLITUDE SPECTRUM OF THE FFT
C     SUMXSQ=       AN ARRAY CONTAINING THE SUM OF THE SQUARES OF THE
C                   VALUES OF THE AMPLITUDE SPECTRUM OF THE FFT
C     SUMX3=        AN ARRAY CONTAINING THE SUM OF THE SQUARES OF THE
C                   VALUES OF THE AMPLITUDE SPECTRUM OF THE FFT
C     SUMX4=        AN ARRAY CONTAINING THE SUM OF THE QUARTICS OF THE
C                   VALUES OF THE AMPLITUDE SPECTRUM OF THE FFT
C     XBAR=         AN ARRAY CONTAINING THE AVERAGE VALUE OF THE
C                   ELEMENTS OF THE AMPLITUDE SPECTRUM
C     VAR=          AN ARRAY CONTAINING THE VARIANCE OF THE VALUES OF
C                   THE ELEMENTS OF THE AMPLITUDE SPECTRUM
C     SKEW=         AN ARRAY CONTAINING THE SKEWNESS OF THE DISTRIBUTION
C                   OF THE ELEMENTS OF THE AMPLITUDE SPECTRUM
C     KUR=          AN ARRAY CONTAINING THE KURTOSIS OF THE DISTRIBUTION
C                   OF THE ELEMENTS OF THE AMPLITUDE SPECTRUM
C     SUMVAR=       SUM OF THE VARIANCES OF AN ELEMENT OF THE AMPLITUDE
C                   SPECTRUM
C     AVAR=         AVERAGE VARIANCE OF AN ELEMENT OF THE AMPLITUDE
C                   SPECTRUM
C     VARVAR=       VARIANCE OF THE VARIANCES OF AN ELEMENT OF THE
C                   AMPLITUDE SPECTRUM
C     SUMVSQ=       SUM OF THE SQUARE OF THE VARIANCES OF AN ELEMENT OF
C                   AMPLITUDE SPECTRUM
C     SUMSKW=       SUM OF THE SKEWNESS OF THE ELEMENT OF THE AMPLITUDE
C                   SPECTRUM
C     ASKEW=        AVERAGE SKEWNESS OF THE ELEMENTS OF THE AMPLITUDE
C                   SPECTRUM
C     VARSKW=       VARIANCE OF THE SKEWNESS OF THE ELEMENTS OF THE
C                   AMPLITUDE SPECTRUM
```

```
C     SUMSSQ=      SUM OF THE SQUARE OF THE SKEWNESS OF THE ELEMENTS OF   STA00610
C                  THE AMPLITUDE SPECTRUM                                 STA00620
C     SUMKUR=      SUM OF THE KURTOSIS OF THE ELEMENTS OF THE AMPLITUDE   STA00630
C                  SPECTRUM                                               STA00640
C     AKUR=        AVERAGE KURTOSIS OF AN ELEMENT OF THE AMPLITUDE        STA00650
C                  SPECTRUM                                               STA00660
C     VARKUR=      VARIANCE OF THE KURTOSIS OF AN ELEMENT OF THE          STA00670
C                  AMPLITUDE SPECTRUM                                     STA00680
C     SUMKSQ=      SUM OF THE SQUARE OF THE KURTOSIS OF THE ELEMENTS OF   STA00690
C                  THE AMPLITUDE SPECTRUM                                 STA00700
C                                                                         STA00710
C     REPDP=       DOUBLE PRECISION VALUE OF REP                          STA00720
C                                                                         STA00730
C     *** VARIABLE DECLARATIONS ***                                       STA00740
C                                                                         STA00750
      INTEGER IN,REP,FLAG,IND2P1                                          STA00760
C                                                                         STA00770
      DOUBLE PRECISION REPDP,SUMVAR,SUMVSQ,AVAR,VARVAR,SUMSKW,SUMSSQ,     STA00780
     *ASKEW,VARSKW,SUMKUR,SUMKSQ,AKUR,VARKUR,NDP                          STA00790
C                                                                         STA00800
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),    STA00810
     *SUMX4(513),XBAR(513),VAR(513),SKEW(513),KUR(513)                    STA00820
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                                STA00830
C                                                                         STA00840
C     *** VARIABLE INITIALIZATION ***                                     STA00850
C                                                                         STA00860
      IND2P1=IN/2+1                                                       STA00870
      IF(REP.EQ.1)THEN                                                    STA00880
         DO 10 I=1,IND2P1                                                 STA00890
              SUMX(I)=0.D0                                                STA00900
              SUMXSQ(I)=0.D0                                              STA00910
              SUMX3(I)=0.D0                                               STA00920
              SUMX4(I)=0.D0                                               STA00930
10       CONTINUE                                                         STA00940
      END IF                                                              STA00950
C                                                                         STA00960
C     *** IF FINAL TIME THROUGH STATS, COMPUTE THE OUTPUT STATISTICS ***STA00970
C                                                                         STA00980
      IF(FLAG.EQ.1)THEN                                                   STA00990
         GO TO 30                                                         STA01000
      END IF                                                              STA01010
C                                                                         STA01020
C     *** ADD THE ELEMENTS OF ARRAY TO THE ELEMENTS OF SUMX, SUMXSQ,      STA01030
C     SUMX3, AND SUMX4 ***                                                STA01040
C                                                                         STA01050
      DO 20 I=1,IND2P1                                                    STA01060
         SUMX(I)=SUMX(I)+ARRAY(I,1)                                       STA01070
         SUMXSQ(I)=SUMXSQ(I)+ARRAY(I,1)**2                                STA01080
         SUMX3(I)=SUMX3(I)+ARRAY(I,1)**3                                  STA01090
         SUMX4(I)=SUMX4(I)+ARRAY(I,1)**4                                  STA01100
```

```
20    CONTINUE
C
C     *** COMPUTE THE FINAL STATISTICS IF FLAG EQUALS 1 ***
C
30    IF(FLAG.EQ. 1)THEN
C
C     *** COMPUTE STATISTICS ASSOCIATED WITH EACH ELEMENT OF FFT ***
C
         REP=REP-1
         REPDP=REP
         DO 40 I=1,IND2P1
            XBAR(I)=SUMX(I)/REP
            IF(REP.EQ.1)THEN
            VAR(I)=((REPDP*SUMXSQ(I))-SUMX(I)**2)/REPDP**2
            ELSE
            VAR(I)=((REPDP*SUMXSQ(I))-SUMX(I)**2)/(REPDP*(REPDP-1.D0))
            END IF
            SKEW(I)=(SUMX3(I)/REPDP)-(3.D0*(SUMXSQ(I)/REPDP)*
     *              (SUMX(I)/REPDP))+(2.D0*(SUMX(I)/REPDP)**2)
            KUR(I)=(SUMX4(I)/REPDP)-(4.D0*(SUMX3(I)/REPDP)*
     *             (SUMX(I)/REPDP))+(6.D0*(SUMX(I)/REPDP)**2*
     *             (SUMXSQ(I)/REPDP))-(3.D0*(SUMX(I)/REPDP)**4)
40       CONTINUE
C
C     *** OUTPUT SOME DATA ***
C
         WRITE(6,50)
50       FORMAT('1',2X,' N',11X,' MEAN',17X,' VARIANCE',17X,' SKEWNESS'
     *   ,16X,' KURTOSIS')
C
         DO 60 I=1,IND2P1
            WRITE(6,70)I,XBAR(I),VAR(I),SKEW(I),KUR(I)
70          FORMAT(1X,I5,2X,E23.16,2X,E23.16,2X,E23.16,2X,E23.16)
60       CONTINUE
C
         WRITE(6,61)
61       FORMAT('1',2X,' N',10X,' SUM X',18X,' SUM X**2',16X,
     *   ' SUM X**3',16X,' SUM X**4')
C
         DO 62 I=1,IND2P1
            WRITE(6,63)I,SUMX(I),SUMXSQ(I),SUMX3(I),SUMX4(I)
63          FORMAT(1X,I5,2X,E23.16,2X,E23.16,2X,E23.16,2X,E23.16)
62       CONTINUE
C
C     *** COMPUTE STATISTICS ASSOCIATED WITH ALL ELEMENTS OF FFT ***
C
         SUMVAR=0.D0
         SUMVSQ=0.D0
         VARVAR=0.D0
         SUMSKW=0.D0
```

```
      SUMSSQ=0.D0                                                       STA01610
      VARSKW=0.D0                                                       STA01620
      SUMKUR=0.D0                                                       STA01630
      SUMKSQ=0.D0                                                       STA01640
      VARKUR=0.D0                                                       STA01650
      NDP=IND2P1                                                        STA01660
C                                                                       STA01670
      DO 80 I=1,IND2P1                                                  STA01680
         SUMVAR=SUMVAR+VAR(I)                                           STA01690
         SUMVSQ=SUMVSQ+(VAR(I)**2)                                      STA01700
         SUMSKW=SUMSKW+SKEW(I)                                          STA01710
         SUMSSQ=SUMSSQ+(SKEW(I)**2)                                     STA01720
         SUMKUR=SUMKUR+KUR(I)                                           STA01730
         SUMKSQ=SUMKSQ+(KUR(I)**2)                                      STA01740
80    CONTINUE                                                          STA01750
C                                                                       STA01760
      AVAR=SUMVAR/NDP                                                   STA01770
      ASKEW=SUMSKW/NDP                                                  STA01780
      AKUR=SUMKUR/NDP                                                   STA01790
C                                                                       STA01800
      DO 90 I=1,IND2P1                                                  STA01810
         VARVAR=VARVAR+(VAR(I)-AVAR)**2                                 STA01820
         VARSKW=VARSKW+(SKEW(I)-ASKEW)**2                               STA01830
         VARKUR=VARKUR+(KUR(I)-AKUR)**2                                 STA01840
90    CONTINUE                                                          STA01850
C                                                                       STA01860
      VARVAR=VARVAR/(NDP-1.D0)                                          STA01870
      VARSKW=VARSKW/(NDP-1.D0)                                          STA01880
      VARKUR=VARKUR/(NDP-1.D0)                                          STA01890
C                                                                       STA01900
      WRITE(6,91)                                                       STA01910
91    FORMAT('1',2X,' SUM OF VARIANCES',7X,' SUM OF VARIANCES**2')      STA01920
C                                                                       STA01930
      WRITE(6,92) SUMVAR,SUMVSQ                                         STA01940
92    FORMAT(1X,E23.16,2X,E23.16)                                       STA01950
C                                                                       STA01960
      WRITE(6,93)                                                       STA01970
93    FORMAT('0',2X,' SUM OF SKEWNESS',9X,' SUM OF SKEWNESS**2')        STA01980
C                                                                       STA01990
      WRITE(6,94) SUMSKW,SUMSSQ                                         STA02000
94    FORMAT(1X,E23.16,2X,E23.16)                                       STA02010
C                                                                       STA02020
      WRITE(6,95)                                                       STA02030
95    FORMAT('0',2X,' SUM OF KURTOSIS',9X,' SUM OF KURTOSIS**2')        STA02040
C                                                                       STA02050
      WRITE(6,96) SUMKUR,SUMKSQ                                         STA02060
96    FORMAT(1X,E23.16,2X,E23.16)                                       STA02070
C                                                                       STA02080
      WRITE(6,100)                                                      STA02090
100   FORMAT('0',3X,' MEAN VARIANCE',7X,' VARIANCE OF THE VARIANCES'STA02100
```

```
     *    )                                                                  STA02110
C                                                                            STA02120
          WRITE(6,101)AVAR,VARVAR                                            STA02130
101       FORMAT(1X,E23.16,2X,E23.16)                                        STA02140
C                                                                            STA02150
          WRITE(6,102)                                                       STA02160
102       FORMAT('0',3X,' MEAN SKEWNESS',7X,' VARIANCE OF THE SKEWNESS')STA02170
C                                                                            STA02180
          WRITE(6,103)ASKEW,VARSKW                                           STA02190
103       FORMAT(1X,E23.16,2X,E23.16)                                        STA02200
C                                                                            STA02210
          WRITE(6,104)                                                       STA02220
104       FORMAT('0',3X,' MEAN KURTOSIS',7X,' VARIANCE OF THE KURTOSIS')STA02230
C                                                                            STA02240
          WRITE(6,105)AKUR,VARKUR                                            STA02250
105       FORMAT(1X,E23.16,2X,E23.16)                                        STA02260
C                                                                            STA02270
          WRITE(6,106)                                                       STA02280
106       FORMAT('1')                                                        STA02290
C                                                                            STA02300
      END IF                                                                 STA02310
C                                                                            STA02320
      RETURN                                                                 STA02330
      END                                                                    STA02340
```

# APPENDIX B

## IMSL/NON-IMSL ROUTINES UTILIZED

### IMSL ROUTINES

```
C   IMSL ROUTINE NAME   - GGUBFS                                         GGU00110
C                                                                        GGU00120
C-----------------------------------------------------------------------GGU00130
C                                                                        GGU00140
C   COMPUTER            - IBM/SINGLE                                     GGU00150
C                                                                        GGU00160
C   LATEST REVISION     - JUNE 1, 1980                                   GGU00170
C                                                                        GGU00180
C   PURPOSE             - BASIC UNIFORM (0,1) RANDOM NUMBER GENERATOR - GGU00190
C                           FUNCTION FORM OF GGUBS                       GGU00200
C                                                                        GGU00210
C   USAGE               - FUNCTION GGUBFS (DSEED)                        GGU00220
C                                                                        GGU00230
C   ARGUMENTS    GGUBFS - RESULTANT DEVIATE.                             GGU00240
C                DSEED  - INPUT/OUTPUT DOUBLE PRECISION VARIABLE         GGU00250
C                           ASSIGNED AN INTEGER VALUE IN THE             GGU00260
C                           EXCLUSIVE RANGE (1.D0, 2147483647.D0).       GGU00270
C                           DSEED IS REPLACED BY A NEW VALUE TO BE       GGU00280
C                           USED IN A SUBSEQUENT CALL.                   GGU00290
C                                                                        GGU00300
C   PRECISION/HARDWARE  - SINGLE/ALL                                     GGU00310
C                                                                        GGU00320
C   REQD. IMSL ROUTINES - NONE REQUIRED                                  GGU00330
C                                                                        GGU00340
C   NOTATION            - INFORMATION ON SPECIAL NOTATION AND            GGU00350
C                           CONVENTIONS IS AVAILABLE IN THE MANUAL       GGU00360
C                           INTRODUCTION OR THROUGH IMSL ROUTINE UHELP   GGU00370
C                                                                        GGU00380
C   COPYRIGHT           - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.        GGU00390
C                                                                        GGU00400
C   WARRANTY            - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN GGU00410
C                           APPLIED TO THIS CODE. NO OTHER WARRANTY,     GGU00420
C                           EXPRESSED OR IMPLIED, IS APPLICABLE.         GGU00430
C                                                                        GGU00440
C-----------------------------------------------------------------------GGU00450
C                                                                        GGU00460
      REAL FUNCTION GGUBFS (DSEED)                                       GGU00470
C                                  SPECIFICATIONS FOR ARGUMENTS          GGU00480
      DOUBLE PRECISION   DSEED                                           GGU00490
C                                  SPECIFICATIONS FOR LOCAL VARIABLES    GGU00500
```

```
      DOUBLE PRECISION   D2P31M,D2P31                                   GGU00510
C                                  D2P31M=(2**31) - 1                   GGU00520
C                                  D2P31 =(2**31) (OR AN ADJUSTED VALUE) GGU00530
      DATA               D2P31M/2147483647.D0/                          GGU00540
      DATA               D2P31 /2147483648.D0/                          GGU00550
C                                  FIRST EXECUTABLE STATEMENT           GGU00560
      DSEED = DMOD(16807.D0*DSEED,D2P31M)                               GGU00570
      GGUBFS = DSEED / D2P31                                            GGU00580
      RETURN                                                            GGU00590
      END                                                               GGU00600
```

```
C   IMSL ROUTINE NAME   - FFTRC                                             FFT00110
C                                                                           FFT00120
C-----------------------------------------------------------------------FFT00130
C                                                                           FFT00140
C   COMPUTER            - IBM/DOUBLE                                        FFT00150
C                                                                           FFT00160
C   LATEST REVISION     - JANUARY 1, 1978                                   FFT00170
C                                                                           FFT00180
C   PURPOSE             - COMPUTE THE FAST FOURIER TRANSFORM OF A           FFT00190
C                           REAL VALUED SEQUENCE                            FFT00200
C                                                                           FFT00210
C   USAGE               - CALL FFTRC (A,N,X,IWK,WK)                         FFT00220
C                                                                           FFT00230
C   ARGUMENTS    A      - INPUT REAL VECTOR OF LENGTH N WHICH               FFT00240
C                           CONTAINS THE DATA TO BE TRANSFORMED.            FFT00250
C                N      - INPUT NUMBER OF DATA POINTS TO BE TRANSFORMED.FFT00260
C                           N MUST BE A POSITIVE EVEN INTEGER.              FFT00270
C                X      - OUTPUT COMPLEX VECTOR OF LENGTH N/2+1             FFT00280
C                           CONTAINING THE FIRST N/2+1 COEFFICIENTS OF      FFT00290
C                           THE FOURIER TRANSFORM. THE REMAINING            FFT00300
C                           COEFFICIENTS MAY BE DETERMINED BY               FFT00310
C                             X(N+2-I) = CONJG(X(I)), FOR I=2,...,N/2.      FFT00320
C                IWK    - INTEGER WORK VECTOR.                              FFT00330
C                           IF N IS A POWER OF 2, THEN IWK SHOULD BE OF     FFT00340
C                           LENGTH M WHERE N=2**M.                          FFT00350
C   PRECISION/HARDWARE  - SINGLE AND DOUBLE/H32                             FFT00430
C                       - SINGLE/H36,H48,H60                                FFT00440
C                                                                           FFT00450
C   REQD. IMSL ROUTINES - FFTCC,FFT2C                                       FFT00460
C                                                                           FFT00470
C   NOTATION            - INFORMATION ON SPECIAL NOTATION AND               FFT00480
C                           CONVENTIONS IS AVAILABLE IN THE MANUAL          FFT00490
C                           INTRODUCTION OR THROUGH IMSL ROUTINE UHELP      FFT00500
C                                                                           FFT00510
C   REMARKS  1.  FFTRC COMPUTES THE FOURIER TRANSFORM, X, ACCORDING         FFT00520
C                TO THE FOLLOWING FORMULA;                                  FFT00530
C                                                                           FFT00540
C                  X(K+1) = SUM FROM J = 0 TO N-1 OF                        FFT00550
C                           A(J+1)*CEXP((0.0,(2.0*PI*J*K)/N))               FFT00560
C                  FOR K=0,1,...,N/2 AND PI=3.1415...                       FFT00570
C                                                                           FFT00580
C                THE USER CAN COMPUTE THE REMAINING X VALUES BY             FFT00590
C                PERFORMING THE FOLLOWING STEPS;                            FFT00600
C                                                                           FFT00610
C                     ND2 = N/2                                             FFT00620
C                     DO 10 I=2,ND2                                         FFT00630
C                        X(N+2-I) = CONJG(X(I))                             FFT00640
C                  10 CONTINUE                                              FFT00650
C            2.  FFTRC CAN BE USED TO COMPUTE                               FFT00660
C                                                                           FFT00670
```

```
C                 X(K+1)  =  (1/N)*SUM FROM J = 0 TO N-1 OF
C                      A(J+1)*CEXP((0.0,(-2.0*PI*J*K)/N))
C                 FOR K=0,1,...,N/2 AND PI=3.1415...
C
C                BY PERFORMING THE FOLLOWING STEPS;
C
C                   CALL FFTRC (A,N,X,IWK,WK)
C                   ND2P1 = N/2+1
C                   DO 10 I=1,ND2P1
C                      X(I) = CONJG(X(I))/N
C                 10 CONTINUE
C
C   COPYRIGHT          - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.
C
C   WARRANTY           - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN
C                          APPLIED TO THIS CODE. NO OTHER WARRANTY,
C                          EXPRESSED OR IMPLIED, IS APPLICABLE.
C
C   MODIFIED 5 AUGUST 1985 TO ALLOW THE GENERATION ONLY OF FFT CF VECTOR
C   OF LENGTH EQUAL TO AN EVEN POWER OF 2 AND THE ENTIRE FFT.
C-----------------------------------------------------------------------
C
      SUBROUTINE FFTRC  (A,N,X,IWK)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            N,IWK(1)
      DOUBLE PRECISION   A(N)
      COMPLEX*16         X(1)
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            ND2P1,ND2,I,MTWO,M,IMAX,ND4,NP2,K,NMK,J
      DOUBLE PRECISION   RPI,ZERO,ONE,HALF,THETA,TP,G(2),B(2),Z(2),AI,
     1                   AR
      COMPLEX*16         XIMAG,ALPH,BETA,GAM,S1,ZD
      EQUIVALENCE        (GAM,G(1)),(ALPH,B(1)),(Z(1),AR),(Z(2),AI),
     1                   (ZD,Z(1))
      DATA               ZERO/0.0D0/,HALF/0.5D0/,ONE/1.0D0/,IMAX/24/
      DATA               RPI/3.141592653589793D0/
C                                  FIRST EXECUTABLE STATEMENT
      IF (N .NE. 2) GO TO 5
C                                  N EQUAL TO 2
      ZD = DCMPLX(A(1),A(2))
      THETA = AR
      TP = AI
      X(2) = DCMPLX(THETA-TP,ZERO)
      X(1) = DCMPLX(THETA+TP,ZERO)
      GO TO 9005
    5 CONTINUE
C                                  N GREATER THAN 2
      ND2 = N/2
      ND2P1 = ND2+1
C                                  MOVE A TO X
```

```
      J = 1                                                             FFT01180
      DO 6  I=1,ND2                                                     FFT01190
         X(I)  = DCMPLX(A(J),A(J+1))                                    FFT01200
         J = J+2                                                        FFT01210
    6 CONTINUE                                                          FFT01220
C                                   COMPUTE THE CENTER COEFFICIENT      FFT01230
      GAM = DCMPLX(ZERO,ZERO)                                           FFT01240
      DO 10 I=1,ND2                                                     FFT01250
         GAM = GAM + X(I)                                               FFT01260
   10 CONTINUE                                                          FFT01270
      TP = G(1)-G(2)                                                    FFT01280
      GAM = DCMPLX(TP,ZERO)                                             FFT01290
C                                   DETERMINE THE SMALLEST M SUCH THAT  FFT01300
C                                   N IS LESS THAN OR EQUAL TO 2**M     FFT01310
      MTWO = 2                                                          FFT01320
      M = 1                                                             FFT01330
      DO 15 I=1,IMAX                                                    FFT01340
         IF (ND2 .LE. MTWO)  GO TO 20                                   FFT01350
         MTWO = MTWO+MTWO                                               FFT01360
         M = M+1                                                        FFT01370
   15 CONTINUE                                                          FFT01380
   20 IF (ND2 .EQ. MTWO)  GO TO 25                                      FFT01390
C                                   N IS NOT A POWER OF TWO, CALL FFTCC FFT01400
C                                   N IS A POWER OF TWO, CALL FFT2C     FFT01410
   25 CALL FFT2C  (X,M,IWK)                                             FFT01420
   30 ALPH = X(1)                                                       FFT01430
      X(1) = B(1)  + B(2)                                               FFT01440
      ND4 = (ND2+1)/2                                                   FFT01450
      IF (ND4 .LT. 2)  GO TO 40                                         FFT01460
      NP2 = ND2 + 2                                                     FFT01470
      THETA = RPI/ND2                                                   FFT01480
      TP = THETA                                                        FFT01490
      XIMAG = DCMPLX(ZERO,ONE)                                          FFT01500
C                                   DECOMPOSE THE COMPLEX VECTOR X      FFT01510
C                                   INTO THE COMPONENTS OF THE TRANSFORM FFT01520
C                                   OF THE INPUT DATA.                  FFT01530
      DO 35 K = 2,ND4                                                   FFT01540
         NMK = NP2 - K                                                  FFT01550
         S1 = DCONJG(X(NMK))                                            FFT01560
         ALPH = X(K)  + S1                                              FFT01570
         BETA = XIMAG*(S1-X(K))                                         FFT01580
         S1 = DCMPLX(DCOS(THETA),DSIN(THETA))                           FFT01590
         X(K)  =  (ALPH+BETA*S1)*HALF                                   FFT01600
         X(NMK) =  DCONJG(ALPH-BETA*S1)*HALF                            FFT01610
         THETA = THETA + TP                                             FFT01620
   35 CONTINUE                                                          FFT01630
   40 CONTINUE                                                          FFT01640
      X(ND2P1) = GAM                                                    FFT01650
C     ND2=N/2                                                           FFT01660
C     DO 90 I=2,ND2                                                     FFT01670
```

```
C         X(N+2-I)=DCONJG(X(I))                                       FFT01680
C90    CONTINUE                                                       FFT01690
 9005 RETURN                                                          FFT01700
      END                                                             FFT01710
```

```
C   IMSL ROUTINE NAME   - FFT2C                                           FFT00110
C                                                                         FFT00120
C-----------------------------------------------------------------------FFT00130
C                                                                         FFT00140
C   COMPUTER            - IBM/DOUBLE                                      FFT00150
C                                                                         FFT00160
C   LATEST REVISION     - NOVEMBER 1, 1984                                FFT00170
C                                                                         FFT00180
C   PURPOSE             - COMPUTE THE FAST FOURIER TRANSFORM OF A         FFT00190
C                           COMPLEX VALUED SEQUENCE OF LENGTH EQUAL TO    FFT00200
C                           A POWER OF TWO                                FFT00210
C                                                                         FFT00220
C   USAGE               - CALL FFT2C (A,M,IWK)                            FFT00230
C                                                                         FFT00240
C   ARGUMENTS    A      - COMPLEX VECTOR OF LENGTH N, WHERE N=2**M.       FFT00250
C                           ON INPUT A CONTAINS THE COMPLEX VALUED        FFT00260
C                           SEQUENCE TO BE TRANSFORMED.                   FFT00270
C                           ON OUTPUT A IS REPLACED BY THE                FFT00280
C                           FOURIER TRANSFORM.                            FFT00290
C                M      - INPUT EXPONENT TO WHICH 2 IS RAISED TO          FFT00300
C                           PRODUCE THE NUMBER OF DATA POINTS, N          FFT00310
C                           (I.E. N = 2**M).                              FFT00320
C                IWK    - WORK VECTOR OF LENGTH M+1.                      FFT00330
C                                                                         FFT00340
C   PRECISION/HARDWARE  - SINGLE AND DOUBLE/H32                           FFT00350
C                       - SINGLE/H36,H48,H60                              FFT00360
C                                                                         FFT00370
C   REQD. IMSL ROUTINES - NONE REQUIRED                                   FFT00380
C                                                                         FFT00390
C   NOTATION            - INFORMATION ON SPECIAL NOTATION AND             FFT00400
C                           CONVENTIONS IS AVAILABLE IN THE MANUAL        FFT00410
C                           INTRODUCTION OR THROUGH IMSL ROUTINE UHELP    FFT00420
C                                                                         FFT00430
C   REMARKS  1.  FFT2C COMPUTES THE FOURIER TRANSFORM, X, ACCORDING       FFT00440
C                TO THE FOLLOWING FORMULA;                                FFT00450
C                                                                         FFT00460
C                  X(K+1) = SUM FROM J = 0 TO N-1 OF                      FFT00470
C                           A(J+1)*CEXP((0.0,(2.0*PI*J*K)/N))             FFT00480
C                  FOR K=0,1,...,N-1 AND PI=3.1415...                     FFT00490
C                                                                         FFT00500
C                NOTE THAT X OVERWRITES A ON OUTPUT.                      FFT00510
C            2.  FFT2C CAN BE USED TO COMPUTE THE INVERSE FOURIER         FFT00520
C                TRANSFORM, X, ACCORDING TO THE FOLLOWING FORMULA;        FFT00530
C                                                                         FFT00540
C                  X(K+1) = (1/N)*SUM FROM J = 0 TO N-1 OF                FFT00550
C                           A(J+1)*CEXP((0.0,(-2.0*PI*J*K)/N))            FFT00560
C                  FOR K=0,1,...,N-1 AND PI=3.1415...                     FFT00570
C                                                                         FFT00580
C                BY PERFORMING THE FOLLOWING STEPS;                       FFT00590
C                                                                         FFT00600
```

```
C                 DO 10 I=1,N                                                  FFT00610
C                    A(I) = CONJG(A(I))                                        FFT00620
C              10 CONTINUE                                                     FFT00630
C                 CALL FFT2C (A,M,IWK)                                         FFT00640
C                 DO 20 I=1,N                                                  FFT00650
C                    A(I) = CONJG(A(I))/N                                      FFT00660
C              20 CONTINUE                                                     FFT00670
C                                                                              FFT00680
C   COPYRIGHT           - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.             FFT00690
C                                                                              FFT00700
C   WARRANTY            - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN       FFT00710
C                           APPLIED TO THIS CODE. NO OTHER WARRANTY,           FFT00720
C                           EXPRESSED OR IMPLIED, IS APPLICABLE.               FFT00730
C                                                                              FFT00740
C-----------------------------------------------------------------------------FFT00750
C                                                                              FFT00760
      SUBROUTINE  FFT2C (A,M,IWK)                                              FFT00770
C                                  SPECIFICATIONS FOR ARGUMENTS                FFT00780
      INTEGER            M,IWK(1)                                              FFT00790
      COMPLEX*16         A(1)                                                  FFT00800
C                                  SPECIFICATIONS FOR LOCAL VARIABLES          FFT00810
      INTEGER            I,ISP,J,JJ,JSP,K,K0,K1,K2,K3,KB,KN,MK,MM,MP,N,        FFT00820
     1                   N4,N8,N2,LM,NN,JK                                     FFT00830
      DOUBLE PRECISION   RAD,C1,C2,C3,S1,S2,S3,CK,SK,SQ,A0,A1,A2,A3,           FFT00840
     1                   B0,B1,B2,B3,TWOPI,TEMP,                               FFT00850
     2                   ZERO,ONE,Z0(2),Z1(2),Z2(2),Z3(2)                      FFT00860
      COMPLEX*16         ZA0,ZA1,ZA2,ZA3,AK2                                   FFT00870
      EQUIVALENCE        (ZA0,Z0(1)),(ZA1,Z1(1)),(ZA2,Z2(1)),                  FFT00880
     1                   (ZA3,Z3(1)),(A0,Z0(1)),(B0,Z0(2)),(A1,Z1(1)),         FFT00890
     2                   (B1,Z1(2)),(A2,Z2(1)),(B2,Z2(2)),(A3,Z3(1)),          FFT00900
     3                   (B3,Z3(2))                                            FFT00910
      DATA               SQ/.70710678118654750D0/,                             FFT00920
     1                   SK/.38268343236508980D0/,                             FFT00930
     2                   CK/.92387953251128680D0/,                             FFT00940
     3                   TWOPI/6.2831853071795860D0/                           FFT00950
      DATA               ZERO/0.0D0/,ONE/1.0D0/                                FFT00960
C                                  SQ=SQRT2/2,SK=SIN(PI/8),CK=COS(PI/8)        FFT00970
C                                  TWOPI=2*PI                                  FFT00980
C                                  FIRST EXECUTABLE STATEMENT                  FFT00990
      MP = M+1                                                                 FFT01000
      N = 2**M                                                                 FFT01010
      IWK(1) = 1                                                               FFT01020
      MM = (M/2)*2                                                             FFT01030
      KN = N+1                                                                 FFT01040
C                                  INITIALIZE WORK VECTOR                      FFT01050
      DO 5  I=2,MP                                                             FFT01060
         IWK(I) = IWK(I-1)+IWK(I-1)                                            FFT01070
    5 CONTINUE                                                                 FFT01080
      RAD = TWOPI/N                                                            FFT01090
      MK = M - 4                                                               FFT01100
```

```
      KB = 1                                                            FFT01110
      IF (MM .EQ. M) GO TO 15                                           FFT01120
      K2 = KN                                                           FFT01130
      K0 = IWK(MM+1) + KB                                               FFT01140
   10 K2 = K2 - 1                                                       FFT01150
      K0 = K0 - 1                                                       FFT01160
      AK2 = A(K2)                                                       FFT01170
      A(K2) = A(K0) - AK2                                               FFT01180
      A(K0) = A(K0) + AK2                                               FFT01190
      IF (K0 .GT. KB) GO TO 10                                          FFT01200
   15 C1 = ONE                                                          FFT01210
      S1 = ZERO                                                         FFT01220
      JJ = 0                                                            FFT01230
      K = MM - 1                                                        FFT01240
      J = 4                                                             FFT01250
      IF (K .GE. 1) GO TO 30                                            FFT01260
      GO TO 70                                                          FFT01270
   20 IF (IWK(J) .GT. JJ) GO TO 25                                      FFT01280
      JJ = JJ - IWK(J)                                                  FFT01290
      J = J-1                                                           FFT01300
      IF (IWK(J) .GT. JJ) GO TO 25                                      FFT01310
      JJ = JJ - IWK(J)                                                  FFT01320
      J = J - 1                                                         FFT01330
      K = K + 2                                                         FFT01340
      GO TO 20                                                          FFT01350
   25 JJ = IWK(J) + JJ                                                  FFT01360
      J = 4                                                             FFT01370
   30 ISP = IWK(K)                                                      FFT01380
      IF (JJ .EQ. 0) GO TO 40                                           FFT01390
C                                  RESET TRIGONOMETRIC PARAMETERS       FFT01400
      C2 = JJ * ISP * RAD                                               FFT01410
      C1 = DCOS(C2)                                                     FFT01420
      S1 = DSIN(C2)                                                     FFT01430
   35 C2 = C1 * C1 - S1 * S1                                            FFT01440
      S2 = C1 * (S1 + S1)                                               FFT01450
      C3 = C2 * C1 - S2 * S1                                            FFT01460
      S3 = C2 * S1 + S2 * C1                                            FFT01470
   40 JSP = ISP + KB                                                    FFT01480
C                                  DETERMINE FOURIER COEFFICIENTS       FFT01490
C                                    IN GROUPS OF 4                     FFT01500
      DO 50 I=1,ISP                                                     FFT01510
         K0 = JSP - I                                                   FFT01520
         K1 = K0 + ISP                                                  FFT01530
         K2 = K1 + ISP                                                  FFT01540
         K3 = K2 + ISP                                                  FFT01550
         ZA0 = A(K0)                                                    FFT01560
         ZA1 = A(K1)                                                    FFT01570
         ZA2 = A(K2)                                                    FFT01580
         ZA3 = A(K3)                                                    FFT01590
         IF (S1 .EQ. ZERO) GO TO 45                                     FFT01600
```

```
      TEMP = A1                                                 FFT01610
      A1 = A1 * C1 - B1 * S1                                    FFT01620
      B1 = TEMP * S1 + B1 * C1                                  FFT01630
      TEMP = A2                                                 FFT01640
      A2 = A2 * C2 - B2 * S2                                    FFT01650
      B2 = TEMP * S2 + B2 * C2                                  FFT01660
      TEMP = A3                                                 FFT01670
      A3 = A3 * C3 - B3 * S3                                    FFT01680
      B3 = TEMP * S3 + B3 * C3                                  FFT01690
   45 TEMP = A0 + A2                                            FFT01700
      A2 = A0 - A2                                              FFT01710
      A0 = TEMP                                                 FFT01720
      TEMP = A1 + A3                                            FFT01730
      A3 = A1 - A3                                              FFT01740
      A1 = TEMP                                                 FFT01750
      TEMP = B0 + B2                                            FFT01760
      B2 = B0 - B2                                              FFT01770
      B0 = TEMP                                                 FFT01780
      TEMP = B1 + B3                                            FFT01790
      B3 = B1 - B3                                              FFT01800
      B1 = TEMP                                                 FFT01810
      A(K0) = DCMPLX(A0+A1,B0+B1)                               FFT01820
      A(K1) = DCMPLX(A0-A1,B0-B1)                               FFT01830
      A(K2) = DCMPLX(A2-B3,B2+A3)                               FFT01840
      A(K3) = DCMPLX(A2+B3,B2-A3)                               FFT01850
   50 CONTINUE                                                  FFT01860
      IF (K .LE. 1) GO TO 55                                    FFT01870
      K = K - 2                                                 FFT01880
      GO TO 30                                                  FFT01890
   55 KB = K3 + ISF                                             FFT01900
C                                 CHECK FOR COMPLETION OF FINAL FFT01910
C                                   ITERATION                   FFT01920
      IF (KN .LE. KB) GO TO 70                                  FFT01930
      IF (J .NE. 1) GO TO 60                                    FFT01940
      K = 3                                                     FFT01950
      J = MK                                                    FFT01960
      GO TO 20                                                  FFT01970
   60 J = J - 1                                                 FFT01980
      C2 = C1                                                   FFT01990
      IF (J .NE. 2) GO TO 65                                    FFT02000
      C1 = C1 * CK + S1 * SK                                    FFT02010
      S1 = S1 * CK - C2 * SK                                    FFT02020
      GO TO 35                                                  FFT02030
   65 C1 = (C1 - S1) * SQ                                       FFT02040
      S1 = (C2 + S1) * SQ                                       FFT02050
      GO TO 35                                                  FFT02060
   70 CONTINUE                                                  FFT02070
C                                 PERMUTE THE COMPLEX VECTOR IN FFT02080
C                                   REVERSE BINARY ORDER TO NORMAL FFT02090
C                                   ORDER                       FFT02100
```

```
      IF(M .LE. 1) GO TO 9005                                           FFT02110
      MP = M+1                                                          FFT02120
      JJ = 1                                                            FFT02130
C                                INITIALIZE WORK VECTOR                 FFT02140
      IWK(1) = 1                                                        FFT02150
      DO 75  I = 2,MP                                                   FFT02160
         IWK(I) = IWK(I-1) * 2                                          FFT02170
   75 CONTINUE                                                          FFT02180
      N4 = IWK(MP-2)                                                    FFT02190
      IF (M .GT. 2) N8 = IWK(MP-3)                                      FFT02200
      N2 = IWK(MP-1)                                                    FFT02210
      LM = N2                                                           FFT02220
      NN = IWK(MP) +1                                                   FFT02230
      MP = MP-4                                                         FFT02240
C                                DETERMINE INDICES AND SWITCH A         FFT02250
      J = 2                                                             FFT02260
   80 JK = JJ + N2                                                      FFT02270
      AK2 = A(J)                                                        FFT02280
      A(J) = A(JK)                                                      FFT02290
      A(JK) = AK2                                                       FFT02300
      J = J+1                                                           FFT02310
      IF (JJ .GT. N4) GO TO 85                                          FFT02320
      JJ = JJ + N4                                                      FFT02330
      GO TO 105                                                         FFT02340
   85 JJ = JJ - N4                                                      FFT02350
      IF (JJ .GT. N8) GO TO 90                                          FFT02360
      JJ = JJ + N8                                                      FFT02370
      GO TO 105                                                         FFT02380
   90 JJ = JJ - N8                                                      FFT02390
      K = MP                                                            FFT02400
   95 IF (IWK(K) .GE. JJ) GO TO 100                                     FFT02410
      JJ = JJ - IWK(K)                                                  FFT02420
      K = K - 1                                                         FFT02430
      GO TO 95                                                          FFT02440
  100 JJ = IWK(K) + JJ                                                  FFT02450
  105 IF (JJ .LE. J) GO TO 110                                          FFT02460
      K = NN - J                                                        FFT02470
      JK = NN - JJ                                                      FFT02480
      AK2 = A(J)                                                        FFT02490
      A(J) = A(JJ)                                                      FFT02500
      A(JJ) = AK2                                                       FFT02510
      AK2 = A(K)                                                        FFT02520
      A(K) = A(JK)                                                      FFT02530
      A(JK) = AK2                                                       FFT02540
  110 J = J + 1                                                         FFT02550
C                                CYCLE REPEATED UNTIL LIMITING NUMBER   FFT02560
C                                  OF CHANGES IS ACHIEVED               FFT02570
      IF (J .LE. LM) GO TO 80                                           FFT02580
C                                                                       FFT02590
 9005 RETURN                                                            FFT02600
```

```
      END                                                                FFT02610
```

NON-IMSL ROUTINE

```
C.......................................................................UTP00110
C                                                                       UTP00120
C     SUBROUTINE UTPLOT                                                 UTP00130
C                                                                       UTP00140
C     PURPOSE                                                           UTP00150
C                                                                       UTP00160
C        PRINTS GRAPHS ON THE STANDARD OUTPUT PRINTER                   UTP00170
C                                                                       UTP00180
C     FEATURES                                                          UTP00190
C                                                                       UTP00200
C     1) FULL CONTROL OVER SCALING                                      UTP00210
C     2) ABILITY TO PLOT SINGLE OR DOUBLE PRECISION VECTORS             UTP00220
C                                                                       UTP00230
C     CALLING SEQUENCE                                                  UTP00240
C                                                                       UTP00250
C     CALL UTPLOT(X,Y,N,RANGE,K,MODCUR)                                 UTP00260
C                                                                       UTP00270
C     DESCRIPTION CF ARGUMENTS                                          UTP00280
C                                                                       UTP00290
C     X       VECTOR OF ABSCISSAE                                       UTP00300
C                                                                       UTP00310
C     Y       VECTOR OF ASSOCIATED ORDINATES                            UTP00320
C                                                                       UTP00330
C     N       NUMBER OF (X,Y) PAIRS                                     UTP00340
C                                                                       UTP00350
C     RANGE   4 WORD SCALING VECTOR WHERE                               UTP00360
C             RANGE(1)= MAXIMUM X TO BE PLOTTED                         UTP00370
C             RANGE(2)= MINIMUM X TO BE PLOTTED                         UTP00380
C             RANGE(3)= MAXIMUM Y TO BE PLOTTED                         UTP00390
C             RANGE(4)= MINIMUM Y TO BE PLOTTED                         UTP00400
C                                                                       UTP00410
C             ALL (X,Y) POINTS OUTSIDE THE ABOVE RANGE WILL BE PLOTTED  UTP00420
C             IN THE BORDER OF THE GRAPH.                               UTP00430
C                                                                       UTP00440
C     K       EVERY KTH ELEMENT OF X & Y WILL BE PLOTTED, E.G.,         UTP00450
C             FOR REAL*4 DATA (SINGLE PRECISION) K=1                    UTP00460
C             FOR REAL*8 DATA (DOUBLE PRECISION) K=2.                   UTP00470
C                                                                       UTP00480
C     MODCUR CONTROLS THE NUMBER OF CURVES ON ONE GRAPH                 UTP00490
C             =0 THERE IS ONLY 1 CURVE ON THIS GRAPH                    UTP00500
C             =1 THIS IS THE FIRST OF TWO OR MORE CURVES ON THIS GRAPH  UTP00510
C             =2 THIS IS AN INTERMEDIATE CURVE ON THIS GRAPH            UTP00520
C             =3 THIS IS THE LAST CURVE ON THIS GRAPH                   UTP00530
C                                                                       UTP00540
C     SCALING                                                           UTP00550
C                                                                       UTP00560
C             SCALING IS PERFORMED ONLY ON THE FIRST SET OF POINTS (WHEN UTP00570
```

```
C         MODCUR = 0 OR 1.) ARRAY RANGE IS USED TO SET UP THE SCALE FAC-
C         TORS AND NEED ONLY BE DEFINED FOR THE FIRST CALL TO UTPLOT.
C
C      GRID LABELLING
C
C            THE DATA TO BE GRAPHED WILL BE FIT INTO AN 80 COLUMN BY
C         BY 60 ROW GRID. THE GRID WILL BE LABELLED THUSLY:
C
C         IN THE X DIRECTION (COLUMN-WISE), THERE WILL BE 5 VALUES:
C         THE MAXIMUM, THE MINIMUM, AND 3 INTERMEDIATE AT INCREMENTS OF
C         (RANGE(2)-RANGE(1))/4. FROM THE MINIMUM.
C
C         IN THE Y DIRECTION (ROW-WISE) THERE WILL BE 7 VALUES: THE MAXI-
C         MUM, THE MINIMUM, AND 5 INTERMEDIATE AT INCREMENTS OF
C         (RANGE(4)-RANGE(3))/6. FROM THE MINIMUM.
C
C         IF THE LABELS HAVE A VALUE BETWEEN 1. AND 10**8, THEY WILL BE
C         PRINTED IN AN F11.2 FORMAT, OTHERWISE THEY WILL BE PRINTED IN A
C         1PE10.3 FORMAT.
C
C      PLOTTING
C
C            FOUR CHARACTERS ARE USED FOR PLOTTING CURVES, ".","+","*",
C         AND "X". WHEN MORE THAN 4 CURVES ARE PLOTTED THE CHARACTERS
C         ARE USED REPEATEDLY.  IF A NEW CURVE IS TO BE PLACED IN THE
C         PLOTTING GRID WHERE AN OLD CURVE EXISTS, THE NEW CURVE REPLACES
C         THE OLD ONE. THUS, IF 3 IDENTICAL CURVES ARE PLOTTED, THEY WILL
C         APPEAR AS ONE CURVE COMPOSED OF "*"'S.
C
C      MESSAGES
C
C            UNDER CERTAIN CIRCUMSTANCES, A PLOT WILL NOT BE OUTPUT AND
C         ONE OF THE FOLLOWING MESSAGES WILL BE PRINTED ON THE STANDARD
C         OUTPUT IN PLACE OF THE PLOT.
C
C         "ALL Y-VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN Y
C          WHEN MODCUR=0 OR 1."
C
C         "ALL X VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX AND MIN X
C          WHEN MODCUR=0 OR 1."
C
C         "GRID NOT SETUP WHEN MODCUR LAST 0 OR 1. NO PLOT UNTIL GRID
C          PROPERLY SETUP."
C
C      NOTE
C
C            THE USER IS EXPECTED TO PROVIDE THE NECESSARY CARRIAGE
C         CONTROLS TO PLACE THE GRAPH PROPERLY ON THE PAGE.  BEFORE
C         CALLING UTPLOT THE USER SHOULD ISSUE A PRINT STATEMENT WHICH
C         EJECTS A PAGE SO THAT THE GRAPH WILL BE PLOTTED AT THE TOP
```

```
C        OF THE NEXT PAGE. A TITLE CAN BE PRINTED AT THE BOTTOM OF
C        THE GRAPH BY ISSUING A PRINT STATEMENT RIGHT AFTER CALLING
C        THE SUBROUTINE.
C
C     MAINTENANCE
C        ALTERED FOR WATFIV COMPILATION BY J. FOUST 5/81
C
C.......................................................................
C
      SUBROUTINE UIPLOT (X   ,Y   ,NDATA,RANGE,KKZ,MODCUR)
      DIMENSION GRID(61,81) ,XSCALE(5) ,YSCALE(7)
      DIMENSION X   (1) ,Y   (1) ,RANGE(4)
      INTEGER*2 GRID,BLANK,DOT,XCHAR(4)/1H.,1H+,1H*,1HX/
      DATA DOT,BLANK/Z4B40,Z4040/
      KDATA=NDATA*KKZ
      IF(MODCUR.GT.1) GO TO 444
C
C     GRID IS THE MATRIX USED TO PLOT THE POINTS
C
      IERR=0
      XMAX=RANGE(1)
      XMIN=RANGE(2)
      YMAX=RANGE(3)
      YMIN=RANGE(4)
C
C     CHECKING X AND Y POINTS AND PLOTTING THOSE OUT OF RANGE
C     AT THE MARGIN
C
      DO 30 I=1,KDATA,KKZ
      IF(X(I).GT.XMAX.OR.X(I).LT.XMIN.OR.Y(I).GT.YMAX.OR.Y(I).LT.YMIN)
     1  IERR=IERR+1
      IF(X(I).LE.XMAX) GO TO 205
      X   (I)=XMAX
      GOTO 210
  205 IF(X(I).GE.XMIN) GO TO 210
      X   (I)=XMIN
  210 IF(Y(I).LE.YMAX) GO TO 215
      Y   (I)=YMAX
      GOTO 30
  215 IF(Y(I).GE.YMIN) GO TO 30
      Y   (I)=YMIN
C
   30 CONTINUE
C
C     PLOTTING X AND Y AXIS , IF NECESSARY
C
      JERR=0
      XRANGE=XMAX-XMIN
      YRANGE=YMAX-YMIN
      IF (YRANGE.NE.0.)  GO TO 298
```

```
      IF(YMIN.EQ.0.) GO TO 889                                          UTP01580
      YMIN=0.                                                           UTP01590
      YRANGE=YMAX                                                       UTP01600
      GO TO 299                                                         UTP01610
  298 IF (XRANGE.NE.0.)   GO TO 299                                     UTP01620
      IF(XMIN.EQ.0.) GO TO 887                                          UTP01630
      XMIN=0.                                                           UTP01640
      XRANGE=XMAX                                                       UTP01650
C                                                                       UTP01660
C     BLANKING OUT MATRIX-(GRID)                                        UTP01670
C                                                                       UTP01680
  299 DO 300 I=1,61                                                     UTP01690
      DO 301 JJ=1,81                                                    UTP01700
301   GRID(I,JJ)=BIANK                                                  UTP01710
  300 CONTINUE                                                          UTP01720
      IF(XMAX*XMIN.GE.0.) GO TO 222                                     UTP01730
      IYAXIS=80.*(-XMIN)/XRANGE+1.5                                     UTP01740
      DO 40 I=1,61                                                      UTP01750
   40 GRID(I,IYAXIS)=DOT                                                UTP01760
  222 IF(YMAX*YMIN.GE.0.) GC TO 333                                     UTP01770
      IXAXIS=60.*YMAX/YRANGE+1.5                                        UTP01780
      DO 60 I=1,81                                                      UTP01790
   60 GRID(IXAXIS,I)=DOT                                                UTP01800
C                                                                       UTP01810
C     COMPUTE PROPER SCALE NUMBERS                                      UTP01820
C                                                                       UTP01830
  333 XINCR=XRANGE/4.                                                   UTP01840
      YINCR=YRANGE/6.                                                   UTP01850
      XSCALE(1)=XMAX                                                    UTP01860
      XSCALE(5)=XMIN                                                    UTP01870
      DO 80 I=2,4                                                       UTP01880
      XSCALE(I)=XSCALE(I-1)-XINCR                                       UTP01890
      IF(ABS(XSCALE(I)).LT.1.E-4)  XSCALE(I)=0.                         UTP01900
   80 CONTINUE                                                          UTP01910
      YSCALE(1)=YMAX                                                    UTP01920
      YSCALE(7)=YMIN                                                    UTP01930
      DO 81 I=2,6                                                       UTP01940
      YSCALE(I)=YSCALE(I-1)-YINCR                                       UTP01950
      IF(ABS(YSCALE(I)).LT.1.E-4)  YSCALE(I)=0.                         UTP01960
   81 CONTINUE                                                          UTP01970
      DO 85 II=1,2                                                      UTP01980
      JJ=6-II                                                           UTP01990
      XT=XSCALE(JJ)                                                     UTP02000
      XSCALE(JJ)=XSCALE(II)                                             UTP02010
   85 XSCALE(II)=XT                                                     UTP02020
C                                                                       UTP02030
C                                                                       UTP02040
C     PLACING POINTS IN THEIR PROPER GRID POSITIONS                     UTP02050
C                                                                       UTP02060
  444 IF(MODCUR.LT.2) JSET=0                                            UTP02070
```

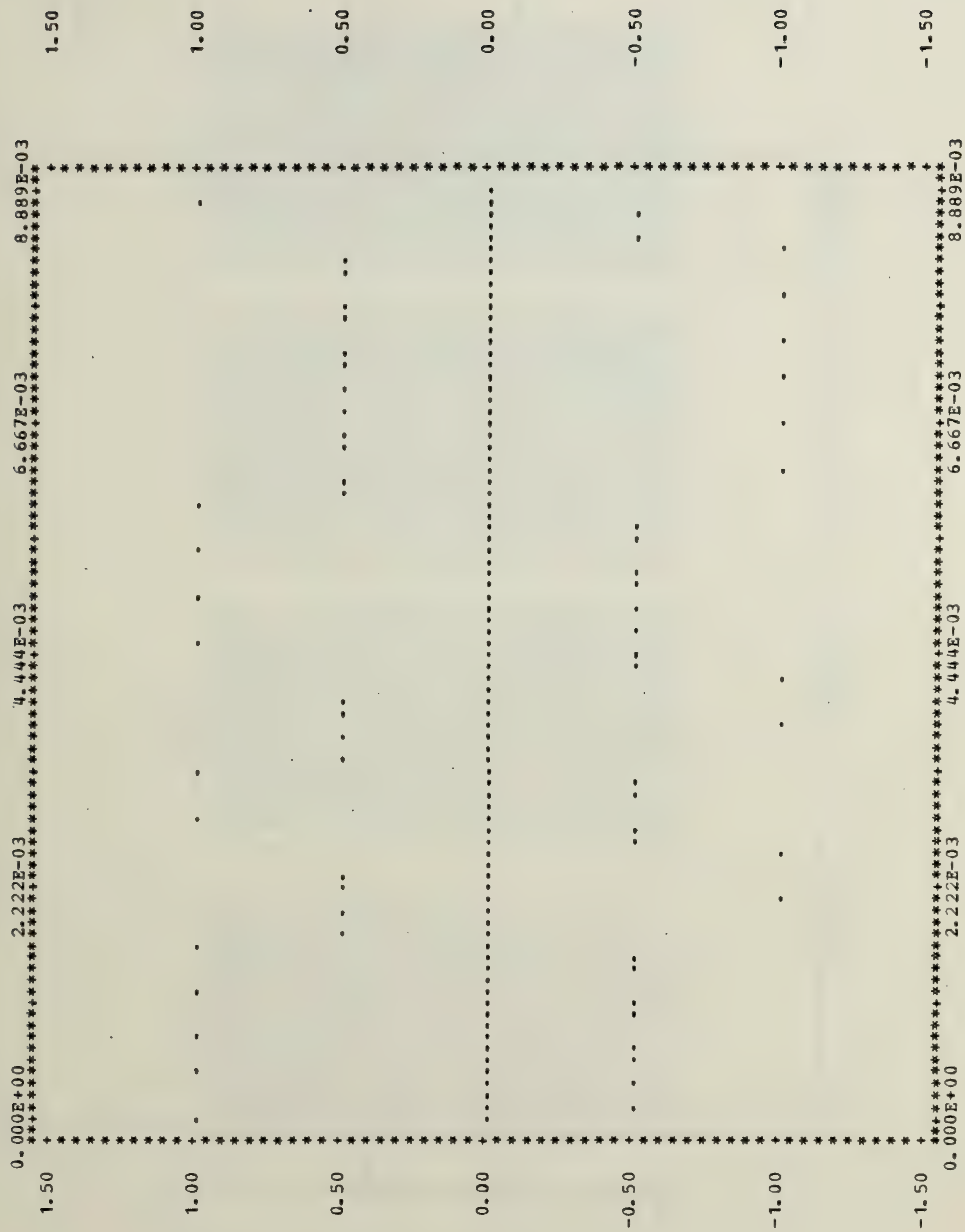
```
      IF(JERR.GT.0) GO TO 885                                          UTP02080
      JSET=JSET+1                                                      UTP02090
      IF(JSET.GT.4) JSET=1                                             UTP02100
      DO 700 I=1,KLATA,KKZ                                             UTP02110
      IPTX=60.*(YMAX-Y(I))/YRANGE+1.5                                  UTP02120
      IPTY=80.*(X (I)-XMIN)/XRANGE+1.5                                 UTP02130
      IF(IPTX.GT.61.OR.IPTY.GT.81) GO TO 70                            UTP02140
      IF(IPTX.LE.0.OR.IPTY.LE.0) GO TO 70                              UTP02150
      GRID(IPTX,IPTY) = XCHAR(JSET)                                    UTP02160
      GO TO 700                                                        UTP02170
   70 IERR=IERR+1                                                      UTP02180
  700 CONTINUE                                                         UTP02190
C     OUTPUT SECTION WITH GRAPH                                        UTP02200
C                                                                      UTP02210
      IF(MODCUR.EQ.1.OR.MODCUR.EQ.2) RETURN                            UTP02220
      AXR=ABS(XRANGE)                                                  UTP02230
      AYR=ABS(YRANGE)                                                  UTP02240
      IF(AXR.LT.1.E+8.AND.AXR.GE..95) GO TO 400                        UTP02250
      WRITE(6,17) XSCALE                                               UTP02260
   17 FORMAT(12X,1PE10.3,4(10X,E10.3)/15X,'**',8('+*********'),'+**')  UTP02270
      GO TO 401                                                        UTP02280
  400 WRITE(6,117) XSCALE                                              UTP02290
  117 FORMAT( 8X,F11.2,4( 9X,F11.2)/15X,'**',8('+*********'),'+**')    UTP02300
  401 II=1                                                             UTP02310
      DO 101 IK=1,61                                                   UTP02320
      IF(MOD(IK-1,10).NE.0) GO TO 92                                   UTP02330
      IF(AYR.LT.1.E+8.AND.AYR.GE..95) GO TO 404                        UTP02340
      WRITE(6,18) YSCALE(II),(GRID(IK,IX),IX=1,81),YSCALE(II)          UTP02350
   18 FORMAT(3X,1PE10.3,2X,1H+,1X,81A1,1X,1H+,2X,E10.3)                UTP02360
      GO TO 405                                                        UTP02370
  404 WRITE(6,118) YSCALE(II),(GRID(IK,IX),IX=1,81),YSCALE(II)         UTP02380
  118 FORMAT(2X,F11.2,'  + ',81A1,' +  ',F11.2)                        UTP02390
  405 II=II+1                                                          UTP02400
      GO TO 101                                                        UTP02410
   92 WRITE(6,19) (GRID(IK,IX),IX=1,81)                                UTP02420
   19 FORMAT(15X,'* ',81A1,' *')                                       UTP02430
  101 CONTINUE                                                         UTP02440
      IF(AXR.LT.1.E+8.AND.AXR.GE..95) GO TO 402                        UTP02450
      WRITE(6,22) XSCALE                                               UTP02460
   22 FORMAT(15X,'**',8('+*********'),'+**'/12X,1PE10.3,4(10X,E10.3),//)UTP02470
      GO TO 403                                                        UTP02480
  402 WRITE(6,217) XSCALE                                              UTP02490
  217 FORMAT(15X,'**',8('+*********'),'+**'/ 8X,F11.2,4( 9X,F11.2),//) UTP02500
  403 IF(IERR.GT.0) WRITE(6,20) IERR                                   UTP02510
   20 FORMAT(10X,'NUMBER OF POINTS OUT OF RANGE =', I4)                UTP02520
 1000 RETURN                                                           UTP02530
C                                                                      UTP02540
  889 WRITE(6,888)                                                     UTP02550
  888 FORMAT(' ALL Y VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN YUTP02560
     1 WHEN MODCUR=0 OR 1.')                                           UTP02570
```

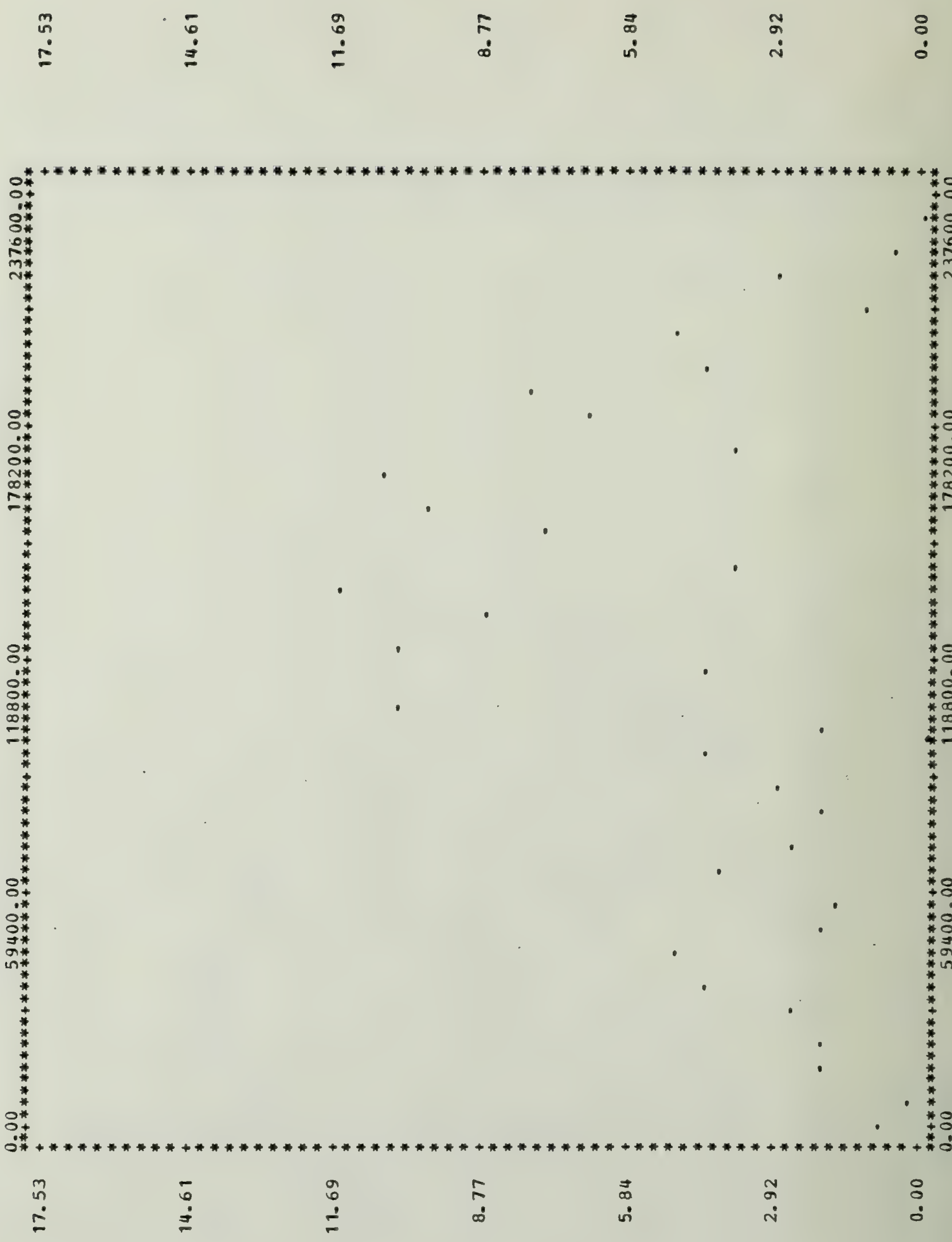
```
      JERR=10                                                           UTP02580
      RETURN                                                            UTP02590
  887 WRITE(6,886)                                                      UTP02600
  886 FORMAT(' ALL X VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN XUTP02610
     1 WHEN MODCUR=0 OR 1.')                                            UTP02620
      JERR=10                                                           UTP02630
      RETURN                                                            UTP02640
  885 WRITE(6,884)                                                      UTP02650
  884 FORMAT(' GRID NOT SETUP WHEN MODCUR LAST 0 OR 1. NO PLOT UNTIL GRIUTP02660
     1D PROPERLY SETUP')                                                UTP02670
      RETURN                                                            UTP02680
      END                                                               UTP02690
```

## APPENDIX C

### REPRESENTATIVE RESULTS OF TRIAL SIMULATIONS

```
MODULATION TECHNIQUE = BPSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           1
BIT RATE =  0.120000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.138888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =           64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.201573472326531 2E+03   0.278328162263990 8E+04
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.103929526472373 9E+05   0.105466506583276 5E+08
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.709790182652583 0E+04   0.564546616177089 8E+07
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.610828704019791 4E+01   0.485004060618770 2E+02
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.314937959007193 5E+03   0.227297355110884 4E+06
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.215087934137146 4E+03   0.128712285037349 3E+06
```

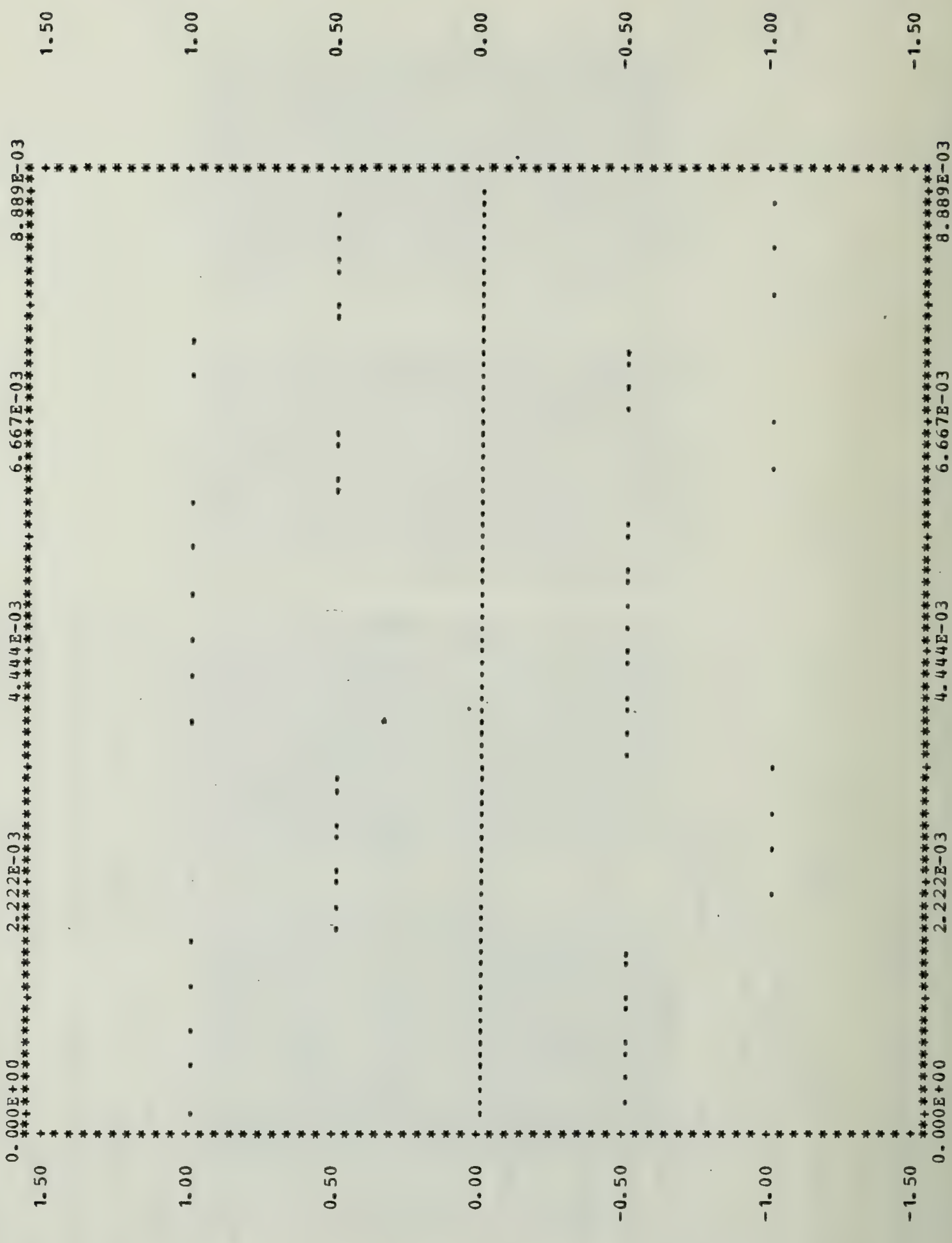

0.000E+00
2.222E-03
4.444E-03
6.667E-03
8.889E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50

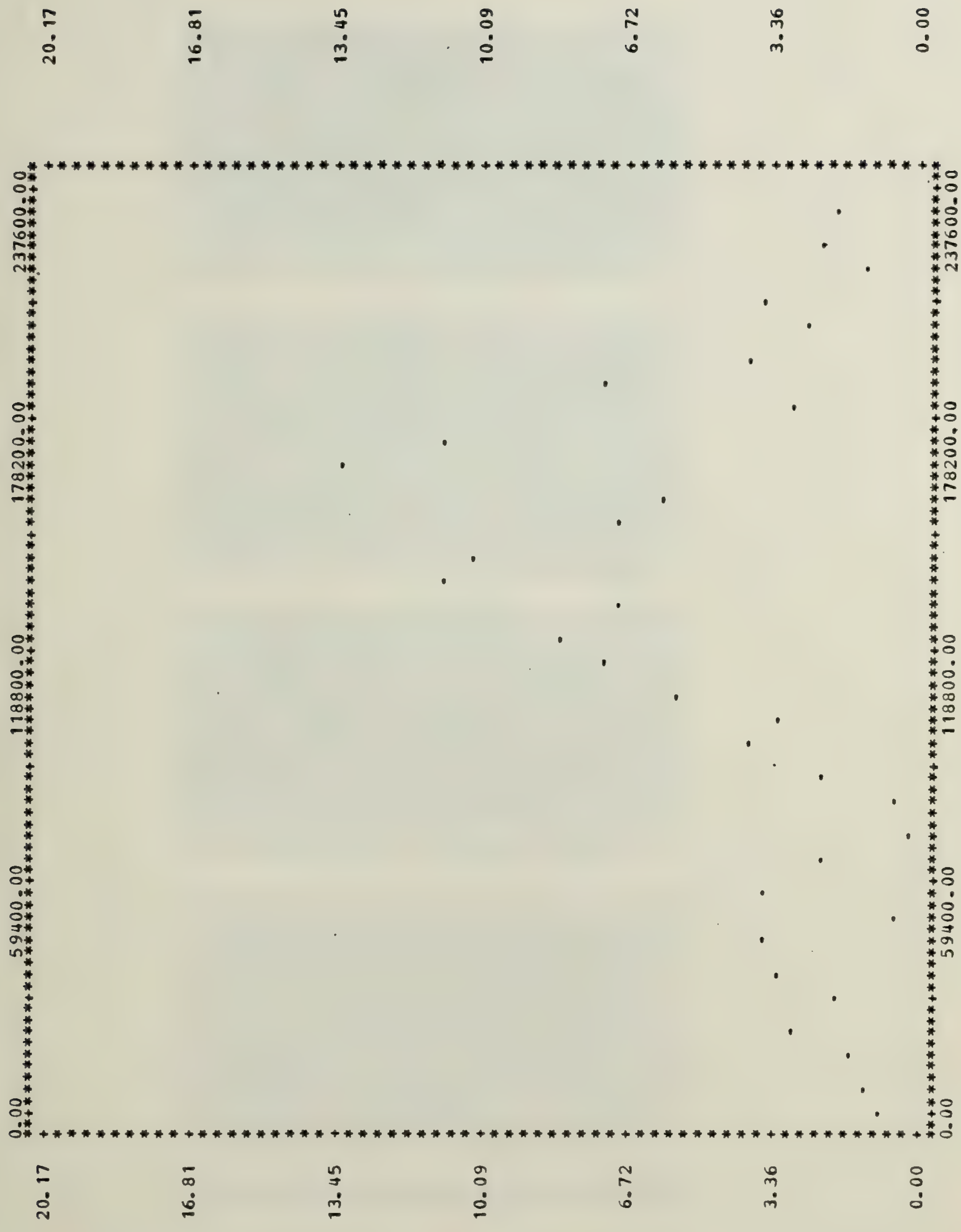

0.00
59400.00
118800.00
178200.00
237600.00
17.53
14.61
11.69
8.77
5.84
2.92
0.00

| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.1000000000000006E+01 | 0.1561758578801998E-14 | -0.1465494392505207E-13 | 0.2220446049250313E-14 |
| 2 | 0.1047493608738669E+01 | 0.2709840268498855E+00 | -0.4986164958259542E-01 | 0.1859590336938843E+00 |
| 3 | 0.1781990149502563E+01 | 0.6052571991271320E+00 | -0.4851377411586108E+01 | 0.9402801672503074E+00 |
| 4 | 0.2374133319422179E+01 | 0.1061217311525198E+01 | -0.1559158728654502E+02 | 0.2555698305934559E+01 |
| 5 | 0.2641583690207520E+01 | 0.1327139362401261E+01 | -0.2238636080694528E+02 | 0.4659815636965693E+01 |
| 6 | 0.2634626472178204E+01 | 0.1802009151563938E+01 | -0.2105244372349955E+02 | 0.1043729257485951E+02 |
| 7 | 0.2930178593409705E+01 | 0.1790464995452298E+01 | -0.3180766187327060E+02 | 0.9230331781179999E+01 |
| 8 | 0.2742584859974633E+01 | 0.1444992379222851E+01 | -0.2556107345444832E+02 | 0.5407086307096591E+01 |
| 9 | 0.2852400569556203E+01 | 0.1931984417605075E+01 | -0.2933416809306337E+02 | 0.8404639812274688E+01 |
| 10 | 0.2056852875467849E+01 | 0.9460987670881228E+00 | -0.8572991054816528E+01 | 0.2193728030163435E+01 |
| 11 | 0.1456809184009350E+01 | 0.1098943847046864E+01 | -0.1522616927873501E+01 | 0.2001109408989501E+01 |
| 12 | 0.1445688920057778E+01 | 0.5573480287919284E+00 | -0.1701780327864210E+01 | 0.5079578481182498E+00 |
| 13 | 0.1920945625822921E+01 | 0.8927281915799500E+00 | -0.6589427533799427E+01 | 0.2348172349608227E+01 |
| 14 | 0.3224566286363947E+01 | 0.2099606310260772E+01 | -0.4608799772325335E+02 | 0.9535238158770767E+01 |
| 15 | 0.4083755764208747E+01 | 0.3945243048148941E+01 | -0.1014772695897762E+03 | 0.3145535439259885E+02 |
| 16 | 0.5391087995993649E+01 | 0.5520139340157494E+01 | -0.2478708803627854E+03 | 0.9424422092077816E+02 |
| 17 | 0.5177315729058215E+01 | 0.1136909276933618E+02 | -0.1938682694723599E+03 | 0.3347284221347290E+03 |
| 18 | 0.7437747541830537E+01 | 0.1296676217270220E+02 | -0.6963060930667482E+03 | 0.4650218840509096E+03 |
| 19 | 0.8281071475316468E+01 | 0.1466781733568776E+02 | -0.9929510812163269E+03 | 0.4278082717480065E+03 |
| 20 | 0.9168033672456426E+01 | 0.2027197304238261E+02 | -0.1367270735397245E+04 | 0.9377448702396778E+03 |
| 21 | 0.8770044710833503E+01 | 0.1938346454362091E+02 | -0.1168636475312538E+04 | 0.9921291201235363E+03 |
| 22 | 0.9524106864593513E+01 | 0.1961533586049662E+02 | -0.1510992880029381E+04 | 0.1378692671328936E+04 |
| 23 | 0.8962219620941568E+01 | 0.1842175759183958E+02 | -0.1262048684336893E+04 | 0.7640763959078340E+03 |
| 24 | 0.7991136976586805E+01 | 0.1591304091880872E+02 | -0.8697872809031544E+03 | 0.5831743233597981E+03 |
| 25 | 0.8132813049012353E+01 | 0.1673585689614668E+02 | -0.9191274726176290E+03 | 0.6136877164277339E+03 |
| 26 | 0.6361858526867442E+01 | 0.8566761867573708E+01 | -0.4268921770206431E+03 | 0.1903785080820935E+03 |
| 27 | 0.5294298817640966E+01 | 0.7006192523842890E+01 | -0.2299556005105339E+03 | 0.1330192136653815E+03 |
| 28 | 0.4057930055889589E+01 | 0.4930941126913854E+01 | -0.9519419493004550E+02 | 0.6781882261712292E+02 |
| 29 | 0.3416828880886142E+01 | 0.2501314042785288E+01 | -0.5501999913351484E+02 | 0.1726642370562735E+02 |
| 30 | 0.2867533428301556E+01 | 0.1412648850252024E+01 | -0.3077393481179399E+02 | 0.4150646443966473E+01 |
| 31 | 0.2020303908837564E+01 | 0.1060744882742050E+01 | -0.8050905470433777E+01 | 0.2614532403606947E+01 |
| 32 | 0.1365571743759467E+01 | 0.4519751609420714E+00 | -0.1273765189038258E+01 | 0.4704424385912696E+00 |
| 33 | 0.1080000000000754E+01 | 0.1003636363636323E+01 | -0.3456000000020134E+00 | 0.1012677119999934E+01 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.1000000000000007E+03 | 0.1000000000000015E+03 | 0.1000000000000022E+03 | 0.1000000000000030E+03 |
| 2 | 0.1047493608738669E+03 | 0.1365517046929747E+03 | 0.2046763767568517E+03 | 0.3383849595387993E+03 |
| 3 | 0.1781990149502563E+03 | 0.3774693520060027E+03 | 0.8977044812979923E+03 | 0.2326050692513711E+04 |
| 4 | 0.2374133319422179E+03 | 0.6687114156800523E+03 | 0.2076369626799344E+04 | 0.6889765711506957E+04 |
| 5 | 0.2641583690207520E+03 | 0.8291832361147630E+03 | 0.2936841778974143E+04 | 0.1138916448013025E+05 |
| 6 | 0.2634626472178204E+03 | 0.8725245707950468E+03 | 0.3402833293597278E+04 | 0.1502031925104808E+05 |
| 7 | 0.2930178593409705E+03 | 0.1035850693477425E+04 | 0.4207727078806093E+04 | 0.1899359170922933E+05 |
| 8 | 0.2742584859974633E+03 | 0.8952314169592701E+03 | 0.3305282702701148E+04 | 0.1337153864187625E+05 |
| 9 | 0.2852400569556203E+03 | 0.1004885358263358E+04 | 0.4038352093600026E+04 | 0.1772010346221309E+05 |
| 10 | 0.2056852875467849E+03 | 0.5167281530737599E+03 | 0.1485073506709137E+04 | 0.4690631226818507E+04 |
| 11 | 0.1456809184009350E+03 | 0.3210247407190382E+03 | 0.8262950792109978E+03 | 0.2278514895214681E+04 |
| 12 | 0.1445688920057778E+03 | 0.2641791002081834E+03 | 0.5575810707434266E+03 | 0.1272769524951599E+04 |
| 13 | 0.1920945625822921E+03 | 0.4573833007032363E+03 | 0.1238876179577409E+04 | 0.3712419048853903E+04 |
| 14 | 0.3224566286363947E+03 | 0.1247643798231314E+04 | 0.5380965068146941E+04 | 0.2495637275319738E+05 |
| 15 | 0.4083755764208747E+03 | 0.2058285175937562E+04 | 0.1173346266754204E+05 | 0.7229234279025983E+05 |
| 16 | 0.5391087995993649E+03 | 0.3452876772730274E+04 | 0.2524443357094650E+05 | 0.2050931970507939E+06 |
| 17 | 0.5177315729058215E+03 | 0.3805999999999641E+04 | 0.3436684441547441E+05 | 0.3486209999999403E+06 |
| 18 | 0.7437747541830537E+03 | 0.6815718304698137E+04 | 0.7138614919385783E+05 | 0.8261276298378326E+06 |
| 19 | 0.8281071475316468E+03 | 0.8309728394163094E+04 | 0.9343002724010123E+05 | 0.1129295226467696E+07 |
| 20 | 0.9168033672456426E+03 | 0.1041220947312536E+05 | 0.1328408193392659E+06 | 0.1833740201322952E+07 |
| 21 | 0.8770044710833503E+03 | 0.9610331412820340E+04 | 0.1206027241518287E+06 | 0.1669696517013331E+07 |
| 22 | 0.9524106864593513E+03 | 0.1101277940700889E+05 | 0.1454196535290649E+06 | 0.2152530635514160E+07 |
| 23 | 0.8962219620941568E+03 | 0.9855892054991122E+04 | 0.1227228629308833E+06 | 0.1661508753740510E+07 |
| 24 | 0.7991136976586806E+03 | 0.7961218068819355E+04 | 0.9110717013921001E+05 | 0.1143542851587153E+07 |
| 25 | 0.8132813049012353E+03 | 0.8271114641737083E+04 | 0.9666101038478942E+05 | 0.1235886896537965E+07 |
| 26 | 0.6361858526867442E+03 | 0.4895433816477395E+04 | 0.4264830561897630E+05 | 0.4069483181946355E+06 |
| 27 | 0.5294298817640966E+03 | 0.3496573056907900E+04 | 0.2693422775779996E+05 | 0.2313456000907510E+06 |
| 28 | 0.4057930055889589E+03 | 0.2134842805413680E+04 | 0.1317634959336049E+05 | 0.9107918677629661E+05 |
| 29 | 0.3416828880886142E+03 | 0.1415102050361508E+04 | 0.6668540831626178E+04 | 0.3463189945762253E+05 |
| 30 | 0.2867533428301556E+03 | 0.9621270324176382E+03 | 0.3554851209525683E+04 | 0.1400578999184000E+05 |
| 31 | 0.2020303908837565E+03 | 0.5131765317978972E+03 | 0.1488901535488743E+04 | 0.4723919613838425E+04 |
| 32 | 0.1365571743759467E+03 | 0.2312241596686921E+03 | 0.4469257803795999E+03 | 0.9444076554027791E+03 |
| 33 | 0.1080000000000754E+03 | 0.2160000000001588E+03 | 0.4320000000004758E+03 | 0.8640000000012697E+03 |

```
MODULATION TECHNIQUE = DBPSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           1
BIT RATE =  0.1200000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.1388888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES          SUM OF VARIANCES**2
 0.2149282745632018E+03   0.3048987837599078E+04
   SUM OF SKEWNESS            SUM OF SKEWNESS**2
-0.9719603900974997E+04   0.8598103883304453E+07
   SUM OF KURTOSIS            SUM OF KURTOSIS**2
 0.8665643426712656E+04   0.8293411900734219E+07
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.6512978017066721E+01   0.5153639719133948E+02
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.2945334515446968E+03   0.1792298562094682E+06
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.2625952553549290E+03   0.1880579703838077E+06
```

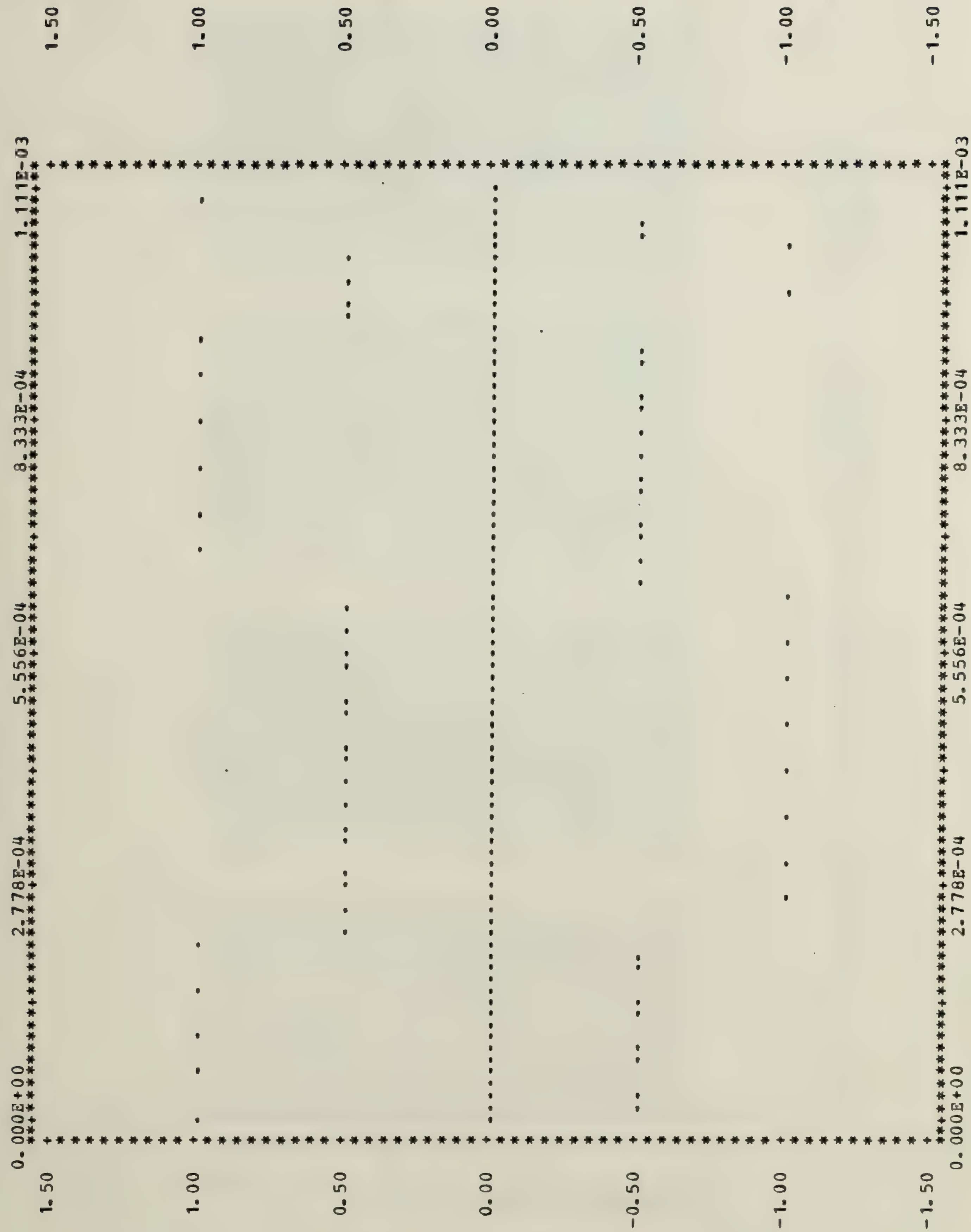

0.000E+00
2.222E-03
4.444E-03
6.667E-03
8.889E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50


0.00
59400.00
118800.00
178200.00
237600.00
20.17
16.81
13.45
10.09
6.72
3.36
0.00

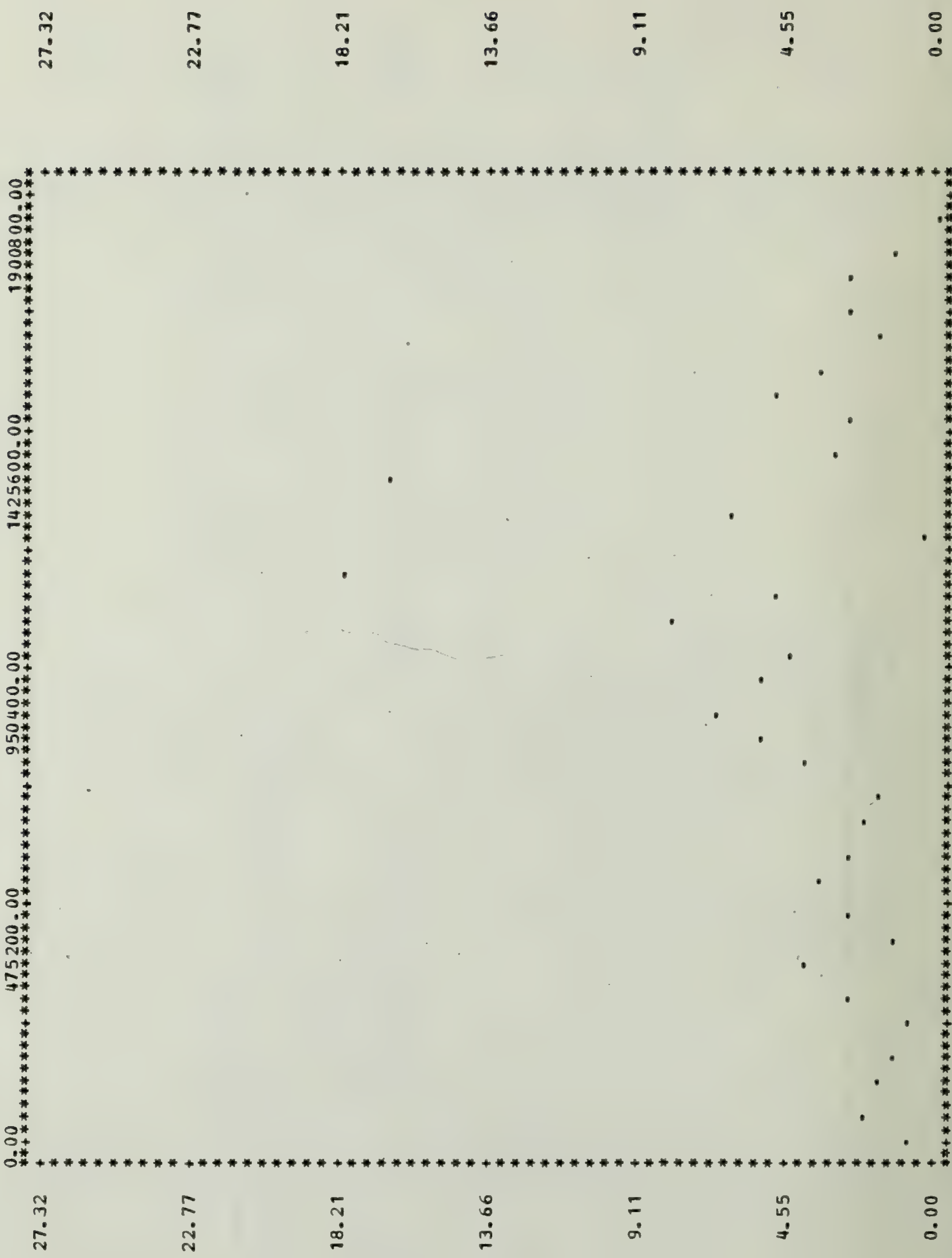
| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.9999999999999964E+00 | 0.1469890427107763E-14 | 0.5773159728050814E-14 | 0.2220446049250313E-14 |
| 2 | 0.1206991215019278E+01 | 0.3394579115190850E+00 | -0.5825378055947565E+00 | 0.2121849517747392E+00 |
| 3 | 0.1614193814360449E+01 | 0.6594138922933482E+00 | -0.3002537615750380E+01 | 0.1142743841392047E+01 |
| 4 | 0.2227669986106446E+01 | 0.1028688761026641E+01 | -0.1153444489528357E+02 | 0.3602977400661302E+01 |
| 5 | 0.2606830384359121E+01 | 0.1860771962726079E+01 | -0.2056467900078679E+02 | 0.8393233099009628E+01 |
| 6 | 0.2869827559993243E+01 | 0.2773607056523804E+01 | -0.2706458826036814E+02 | 0.2484078413586156E+02 |
| 7 | 0.3193892922935319E+01 | 0.1919637658811091E+01 | -0.4389937025518355E+02 | 0.1049069398838691E+02 |
| 8 | 0.2887029562873031E+01 | 0.2188092253886460E+01 | -0.2996273462075921E+02 | 0.1240623718927321E+02 |
| 9 | 0.2645128655136111E+01 | 0.1564423519331396E+01 | -0.2282782127376995E+02 | 0.4783908803836340E+01 |
| 10 | 0.2165532638363491E+01 | 0.1012907097513055E+01 | -0.1023789451958566E+02 | 0.3065360931783477E+01 |
| 11 | 0.1691277358074538E+01 | 0.1004960153591563E+01 | -0.3911484804814908E+01 | 0.1499184701566339E+01 |
| 12 | 0.1589018609218827E+01 | 0.6409517041368564E+00 | -0.2957243584462853E+01 | 0.6008947870243588E+00 |
| 13 | 0.2022206621896048E+01 | 0.8674855068396650E+00 | -0.8124920456579111E+01 | 0.1759584375414942E+01 |
| 14 | 0.2893180879486128E+01 | 0.2056527174777121E+01 | -0.3029972471223451E+02 | 0.1099295459446458E+02 |
| 15 | 0.3800440008063125E+01 | 0.3850369283893350E+01 | -0.7769874123112918E+02 | 0.3737869876870587E+02 |
| 16 | 0.5570047561752625E+01 | 0.7494036582454018E+01 | -0.2769226081575563E+03 | 0.1304851489126716E+03 |
| 17 | 0.6309396047434741E+01 | 0.1987022375415137E+02 | -0.3359332752853699E+03 | 0.1232319292891567E+04 |
| 18 | 0.7725994558988540E+01 | 0.1531549511531786E+02 | -0.7877268097177533E+03 | 0.5389886730055789E+03 |
| 19 | 0.7809215803380767E+01 | 0.1544649313954077E+02 | -0.8025080297305087E+03 | 0.7311368578778502E+03 |
| 20 | 0.8414654985868134E+01 | 0.1653486279572041E+02 | -0.1010517011328395E+04 | 0.8342365979814513E+03 |
| 21 | 0.9176381706965888E+01 | 0.2299461744077966E+02 | -0.1317751057163412E+04 | 0.1483070263595537E+04 |
| 22 | 0.8737450979982724E+01 | 0.2048224307443957E+02 | -0.1127935351642733E+04 | 0.1450198162098086E+04 |
| 23 | 0.8709795816472441E+01 | 0.1666453081075322E+02 | -0.1160347310188729E+04 | 0.6012382067100471E+03 |
| 24 | 0.8359446618565224E+01 | 0.1416732832613374E+02 | -0.1023422594961601E+04 | 0.5124736984154824E+03 |
| 25 | 0.7365872951243031E+01 | 0.1262467755534349E+02 | -0.6663865046499641E+03 | 0.4404715992828096E+03 |
| 26 | 0.6265420185247174E+01 | 0.1029341182529051E+02 | -0.3942727760614418E+03 | 0.2735691108412666E+03 |
| 27 | 0.5661086207993509E+01 | 0.7478565244377920E+01 | -0.2891508150107528E+03 | 0.1483181453821855E+03 |
| 28 | 0.4637040646258831E+01 | 0.6384887129770313E+01 | -0.1450576153004111E+03 | 0.1243505589276638E+03 |
| 29 | 0.3580536036274782E+01 | 0.3579040429595400E+01 | -0.6316134998462517E+02 | 0.3454042237282812E+02 |
| 30 | 0.2568303570148102E+01 | 0.1367376958163926E+01 | -0.2008704187963902E+02 | 0.5364378040649946E+01 |
| 31 | 0.1791906790941165E+01 | 0.8411220788771688E+00 | -0.4841294269761784E+01 | 0.1786012611331170E+01 |
| 32 | 0.1390475344573220E+01 | 0.6184320019866056E+00 | -0.1208132606045036E+01 | 0.9141790760984658E+00 |
| 33 | 0.9200000000007791E+00 | 0.1003636363636374E+01 | 0.2943999999989106E+00 | 0.1012677120000026E+01 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.9999999999999964E+02 | 0.9999999999999943E+02 | 0.9999999999999921E+02 | 0.9999999999999902E+02 |
| 2 | 0.1206991215019278E+03 | 0.1792891125537607E+03 | 0.2995818122167570E+03 | 0.5371329498846672E+03 |
| 3 | 0.1614193814360449E+03 | 0.3258441423689951E+03 | 0.7565487015339066E+03 | 0.1941759256884656E+04 |
| 4 | 0.2227669986106446E+03 | 0.5980915440415869E+03 | 0.1851104541888258E+04 | 0.6434635717617218E+04 |
| 5 | 0.2606830384359121E+03 | 0.8637728895916740E+03 | 0.3339547410677492E+04 | 0.1429682053630766E+05 |
| 6 | 0.2869827559993244E+03 | 0.1098178121005534E+04 | 0.5101104641473661E+04 | 0.2712332783742979E+05 |
| 7 | 0.3193892922935319E+03 | 0.1210139328539929E+04 | 0.5165038885414519E+04 | 0.2418578125586438E+05 |
| 8 | 0.2887029562873032E+03 | 0.1050115102825044E+04 | 0.4431878637369578E+04 | 0.2074597188371301E+05 |
| 9 | 0.2645128655136111E+03 | 0.8545484886360252E+03 | 0.3099048835661787E+04 | 0.1207994399427354E+05 |
| 10 | 0.2165532638363491E+03 | 0.5692309634355466E+03 | 0.1736368916738238E+04 | 0.5928142778063767E+04 |
| 11 | 0.1691277358074538E+03 | 0.3855329653991237E+03 | 0.9928972246440080E+03 | 0.2704860929592714E+04 |
| 12 | 0.1589018609218827E+03 | 0.3159522327539222E+03 | 0.7054415458756292E+03 | 0.1669947189847840E+04 |
| 13 | 0.2022206621896049E+03 | 0.4948130273411497E+03 | 0.1371486571483153E+04 | 0.4145744104203795E+04 |
| 14 | 0.2893180879486128E+03 | 0.1040645750445347E+04 | 0.4328257571012639E+04 | 0.1994425856680307E+05 |
| 15 | 0.3800440008063125E+03 | 0.1825520984594126E+04 | 0.1015480598214063E+05 | 0.6249208342092874E+05 |
| 16 | 0.5570047561752626E+03 | 0.3844452605681585E+04 | 0.3034400480385781E+05 | 0.2622371362589436E+06 |
| 17 | 0.6309396047434741E+03 | 0.5947999999999499E+04 | 0.7102983984520202E+05 | 0.9705829999998300E+06 |
| 18 | 0.7725994558988540E+03 | 0.7485333208968521E+04 | 0.8278405157724163E+05 | 0.1000317965825323E+07 |
| 19 | 0.7809215803380767E+03 | 0.7627587967191728E+04 | 0.8624886821940661E+05 | 0.1092008676776739E+07 |
| 20 | 0.8414654985868135E+03 | 0.8717593269895867E+04 | 0.1048536341808207E+06 | 0.1413147608028859E+07 |
| 21 | 0.9176381706965888E+03 | 0.1069706524983101E+05 | 0.1458647596675802E+06 | 0.2225002804163320E+07 |
| 22 | 0.8737450979982725E+03 | 0.9662047027129625E+04 | 0.1252028417077052E+06 | 0.1843532137866726E+07 |
| 23 | 0.8709795816472441E+03 | 0.9235842866728653E+04 | 0.1101200770348890E+06 | 0.1419247629793849E+07 |
| 24 | 0.8359446618565224E+03 | 0.8390600281151395E+04 | 0.9410399639412192E+05 | 0.1144827215115574E+07 |
| 25 | 0.7365872951243031E+03 | 0.6675451511364377E+04 | 0.7002171584291269E+05 | 0.8171450560058780E+06 |
| 26 | 0.6265420185247174E+03 | 0.4944596780474033E+04 | 0.4566155380318585E+05 | 0.4693948257621125E+06 |
| 27 | 0.5661086207993509E+03 | 0.3945167664626847E+04 | 0.3167714185138088E+05 | 0.2816542865255081E+06 |
| 28 | 0.4637040646258831E+03 | 0.2782318421352912E+04 | 0.1989898011089206E+05 | 0.1612723577315059E+06 |
| 29 | 0.3580536036274782E+03 | 0.1636348833236177E+04 | 0.8696835236079992E+04 | 0.5144863119613108E+05 |
| 30 | 0.2568303570148102E+03 | 0.7949886417017775E+03 | 0.2797375666478596E+04 | 0.1086402293290285E+05 |
| 31 | 0.1791906790941165E+03 | 0.4043640805509461E+03 | 0.1047432809395370E+04 | 0.2988922040732839E+04 |
| 32 | 0.1390475344573220E+03 | 0.2545669365832754E+03 | 0.5544095492100302E+03 | 0.1343313554680140E+04 |
| 33 | 0.9200000000007791E+02 | 0.1840000000001444E+03 | 0.3680000000004329E+03 | 0.7360000000011546E+03 |

```
MODULATION TECHNIQUE = ORTHOGONAL BPSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =            3
BIT RATE =  0.9600000000000000E+04
CARRIER FREQUENCY =       19200 HZ
MAXIMUM CARRIER AMPLITUDE =            1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.2450980392156863E-04 SEC
NUMBER OF SAMPLES GENERATED =           64
SEED FOR RANDOM NUMBER GENERATOR =            1
NUMBER OF TIMES SIMULATION REPEATS =          100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.3565282677437312E+03   0.1043543362358570E+05
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.5381895541377615E+04   0.3582631135045827E+07
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.3658999158877697E+05   0.3330606974397919E+09
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.1080388690132519E+02   0.2057357044299072E+03
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.1630877436781095E+03   0.8452843545285224E+05
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.1108787623902332E+04   0.9140317737733299E+07
```



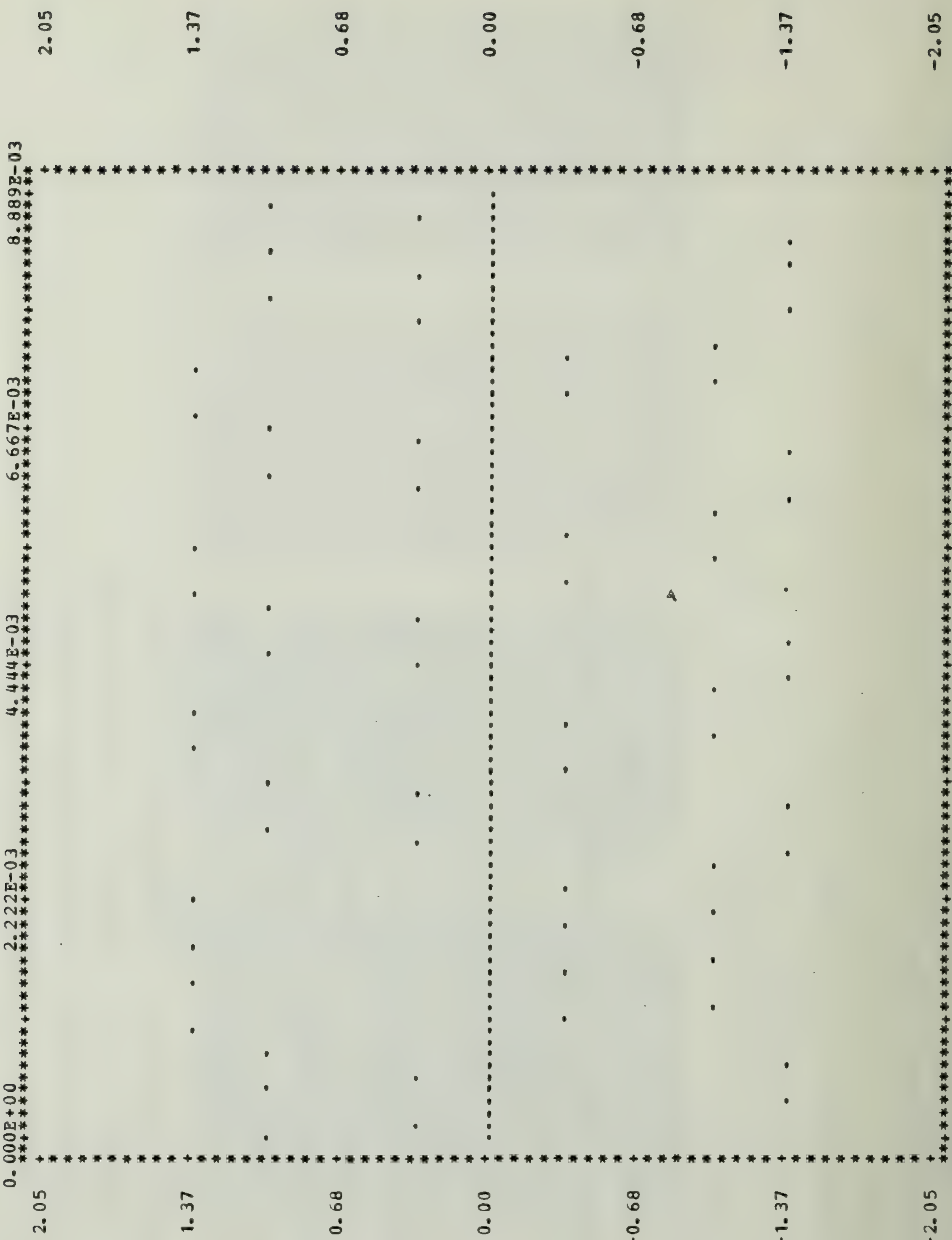
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50
0.000E+00
2.778E-04
5.556E-04
8.333E-04
1.111E-03


0.00
475200.00
950400.00
1425600.00
1900800.00
27.32
22.77
18.21
13.66
9.11
4.55
0.00

| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.9999999999999837E+00 | 0.1194285972025057E-14 | 0.3197442310920451E-13 | 0.1110223024625157E-14 |
| 2 | 0.1411555067139362E+01 | 0.2075712540037567E+00 | -0.1555104112983186E+01 | 0.1477828163804993E+00 |
| 3 | 0.1581662562971217E+01 | 0.6913486002106669E+00 | -0.2512033644738964E+01 | 0.1326654635257686E+01 |
| 4 | 0.1746011668828425E+01 | 0.1660728937560371E+01 | -0.2735720817539909E+01 | 0.7181153218390413E+01 |
| 5 | 0.2487376832401961E+01 | 0.3187132562789080E+01 | -0.1345866539145792E+02 | 0.2313107027830058E+02 |
| 6 | 0.2392317133198317E+01 | 0.4403999674859614E+01 | -0.1579140068729958E+00 | 0.9523892208727593E+02 |
| 7 | 0.2873941993780831E+01 | 0.3317611212361697E+01 | -0.2808951707223596E+02 | 0.2224684476840443E+02 |
| 8 | 0.2936029392917652E+01 | 0.3699996587518791E+01 | -0.2980126647160134E+02 | 0.2984697963609836E+02 |
| 9 | 0.2478264041916265E+01 | 0.2971723315233370E+01 | -0.1211008099582720E+02 | 0.2739050343838217E+02 |
| 10 | 0.1775820899350969E+01 | 0.1041075605053106E+01 | -0.4595670096970628E+01 | 0.2408205984396261E+01 |
| 11 | 0.1518861217105241E+01 | 0.8301259957717636E+00 | -0.2277490279198786E+01 | 0.1192379894500322E+01 |
| 12 | 0.1434317744204484E+01 | 0.5944479879864332E+00 | -0.1796914098729745E+01 | 0.4293200636053687E+00 |
| 13 | 0.1945015183250301E+01 | 0.4916250157649399E+00 | -0.6735034995725021E+01 | 0.1041129156195698E+01 |
| 14 | 0.2804658738339067E+01 | 0.2804358963681170E+01 | -0.2443773332277449E+02 | 0.2289686391698299E+02 |
| 15 | 0.3923050852610531E+01 | 0.3074810213843398E+01 | -0.8866410939161910E+02 | 0.2115406187017686E+02 |
| 16 | 0.5644012086255505E+01 | 0.1119968617493584E+02 | -0.2816643707881831E+03 | 0.2276781264367641E+03 |
| 17 | 0.5090345713940015E+01 | 0.3243270758845300E+02 | 0.1803958842095260E+03 | 0.6547664096250286E+04 |
| 18 | 0.6979480897242355E+01 | 0.1989841039462072E+02 | -0.5293083246138370E+03 | 0.8340250774034976E+03 |
| 19 | 0.7219639256545543E+01. | 0.1405894216350526E+02 | -0.6450669112614341E+03 | 0.4415132116379600E+03 |
| 20 | 0.7426325641963908E+01 | 0.3143013343320532E+02 | -0.5045975981675309E+03 | 0.2873504517215812E+04 |
| 21 | 0.9181074919421593E+01 | 0.3147559988244039E+02 | -0.1221066034803840E+04 | 0.2645319575375468E+04 |
| 22 | 0.7811178351200367E+01 | 0.6501637985261742E+02 | -0.3234257528646465E+02 | 0.1620957998609241E+05 |
| 23 | 0.7155173881115702E+01 | 0.2437036621446784E+02 | -0.5677598575289374E+03 | 0.1100562606554644E+04 |
| 24 | 0.7296060888561213E+01 | 0.1728529342726921E+02 | -0.6196321132882680E+03 | 0.8989413566346584E+03 |
| 25 | C.6686881379771213E+01 | 0.2957759461549051E+02 | -0.2835528998457502E+03 | 0.2878907433833486E+04 |
| 26 | 0.6639869155630594E+01 | 0.1817828764231562E+02 | -0.4549285871737381E+03 | 0.6616207898589500E+03 |
| 27 | 0.4767489656454035E+01 | 0.1017033455698602E+02 | -0.1467728258669405E+03 | 0.2739258324928540E+03 |
| 28 | 0.3442148531187325E+01 | 0.1107312574626655E+02 | 0.6609619933125398E+01 | 0.6323762627354242E+03 |
| 29 | 0.3339390175580569E+01 | 0.6407469821591361E+01 | -0.4267851440326571E+02 | 0.8996253220473250E+02 |
| 30 | 0.2345263541776711E+01 | 0.2296921742748596E+01 | -0.1178274900462600E+02 | 0.1416383521220494E+02 |
| 31 | 0.1936721497866493E+01 | 0.8419979276324613E+00 | -0.6653324231077595E+01 | 0.2421363813672397E+01 |
| 32 | 0.1507494021765924E+01 | 0.8299757840624045E+00 | -0.2320704558100585E+01 | 0.1189920939805388E+01 |
| 33 | 0.9600000000005325E+00 | 0.1008484848484935E+01 | 0.1535999999991078E+00 | 0.1003192320000180E+01 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.9999999999999837E+02 | 0.9999999999999686E+02 | 0.9999999999999536E+02 | 0.9999999999999387E+02 |
| 2 | 0.1411555067139362E+03 | 0.2197983249030528E+03 | 0.3767643649852634E+03 | 0.7054022920508221E+03 |
| 3 | 0.1581662562971217E+03 | 0.3186091577313239E+03 | 0.7602618739154471E+03 | 0.2037758005591990E+04 |
| 4 | 0.1746011668828425E+03 | 0.4692678395869791E+03 | 0.1574757939883318E+04 | 0.5920869916394998E+04 |
| 5 | 0.2487376832401961E+03 | 0.9342304743531201E+03 | 0.4388074473669741E+04 | 0.2277539083259013E+05 |
| 6 | 0.2392317133198317E+03 | 0.1008314094390523E+04 | 0.6076193597121237E+04 | 0.4287047552684854E+05 |
| 7 | 0.2873941993780832E+03 | 0.1154397768385502E+04 | 0.5492156448322934E+04 | 0.2861846863853825E+05 |
| 8 | 0.2936029392917652E+03 | 0.1228326521772000E+04 | 0.6115027949693274E+04 | 0.3356198711125626E+05 |
| 9 | 0.2478264041916265E+03 | 0.9083798743536179E+03 | 0.4314248905159208E+04 | 0.2334844211414864E+05 |
| 10 | 0.1775820899350969E+03 | 0.4184204715574260E+03 | 0.1138844471312494E+04 | 0.3396766130248446E+04 |
| 11 | 0.1518861217105241E+03 | 0.3128764132640459E+03 | 0.7365106422760920E+03 | 0.1859737061213269E+04 |
| 12 | 0.1434317744204484E+03 | 0.2645770899446408E+03 | 0.5473179563118097E+03 | 0.1186913250208475E+04 |
| 13 | 0.1945015183250301E+03 | 0.4269792828681493E+03 | 0.1061323252148284E+04 | 0.2962998661477288E+04 |
| 14 | 0.2804658738339067E+03 | 0.1064242601258605E+04 | 0.4937516474012145E+04 | 0.2601588997665944E+05 |
| 15 | 0.3923050852610531E+03 | 0.1843439010387318E+04 | 0.9751238406710893E+04 | 0.5596569546028266E+05 |
| 16 | 0.5644012086255505E+03 | 0.4294256174298470E+04 | 0.3817308968287571E+05 | 0.3682273043798486E+06 |
| 17 | 0.5090345713940015E+03 | 0.5801999999999595E+04 | 0.1014598220203008E+06 | 0.2020016999999756E+07 |
| 18 | 0.6979480897242356E+03 | 0.6841257988564547E+04 | 0.8057182515250085E+05 | 0.1045136458643242E+07 |
| 19 | 0.7219639256545543E+03 | 0.6604154373652368E+04 | 0.6810750719184861E+05 | 0.7606683646978952E+06 |
| 20 | 0.7426325641963908E+03 | 0.8626614463936393E+04 | 0.1307023222657515E+06 | 0.2227806669826787E+07 |
| 21 | 0.9181074919421593E+03 | 0.1154529805596482E+05 | 0.1790297084410059E+06 | 0.3131754862963641E+07 |
| 22 | 0.7811178351200367E+03 | 0.1253807232883525E+05 | 0.2783741984468482E+06 | 0.6845485247415800E+07 |
| 23 | 0.7155173881115703E+03 | 0.7532317582132351E+04 | 0.9466983767713212E+05 | 0.1292127369194078E+07 |
| 24 | 0.7296060888561213E+03 | 0.7034494498258915E+04 | 0.8136258831189237E+05 | 0.1067727029668344E+07 |
| 25 | 0.6686881379771213E+03 | 0.7399620125646657E+04 | 0.1111429796046996E+06 | 0.1875286557886083E+07 |
| 26 | 0.6639869155630594E+03 | 0.6208436716978700E+04 | 0.6935919118710130E+05 | 0.8491275569515316E+06 |
| 27 | 0.4767489656454035E+03 | 0.3279758883581237E+04 | 0.2768557554783706E+05 | 0.2630640137869949E+06 |
| 28 | 0.3442148531187325E+03 | 0.2281078099955895E+04 | 0.2184671358502179E+05 | 0.2439888675208497E+06 |
| 29 | 0.3339390175580569E+03 | 0.1749492186813947E+04 | 0.1102855427342501E+05 | 0.7656074754229115E+05 |
| 30 | 0.2345263541776711E+03 | 0.7774213605708154E+03 | 0.3191446804095530E+04 | 0.1477525773195163E+05 |
| 31 | 0.1936721497866493E+03 | 0.4584468108654369E+03 | 0.1248140927526872E+04 | 0.3814593155215503E+04 |
| 32 | 0.1507494021765924E+03 | 0.3094214251881780E+03 | 0.7127747450903528E+03 | 0.1747303658994075E+04 |
| 33 | 0.9600000000005325E+02 | 0.1920000000001109E+03 | 0.3840000000003322E+03 | 0.7680000000008864E+03 |

```
MODULATION TECHNIQUE = QPSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =         1200 HZ
BITS PER BINARY CODE WORD =            2
BIT RATE =  0.240000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =            1 VOLT(S)
INITIAL PHASE ANGLE =          0 DEGREES
TIME BETWEEN SAMPLES =  0.138888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =         64
SEED FOR RANDOM NUMBER GENERATOR =          1
NUMBER OF TIMES SIMULATION REPEATS =        100
   SUM OF VARIANCES          SUM OF VARIANCES**2
 0.4090708290354776E+03   0.106291980142703 3E+05
   SUM OF SKEWNESS            SUM OF SKEWNESS**2
-0.286798208534296 9E+05   0.735227091963426 6E+08
   SUM OF KURTOSIS            SUM OF KURTOSIS**2
 0.317213053861798 8E+05   0.975157710572189 1E+08
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.123960857283478 1E+02   0.173697529643132 2E+03
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.869085480406960 4E+03   0.151867166599856 7E+07
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.961251678369087 1E+03   0.209448790670922 3E+07
```

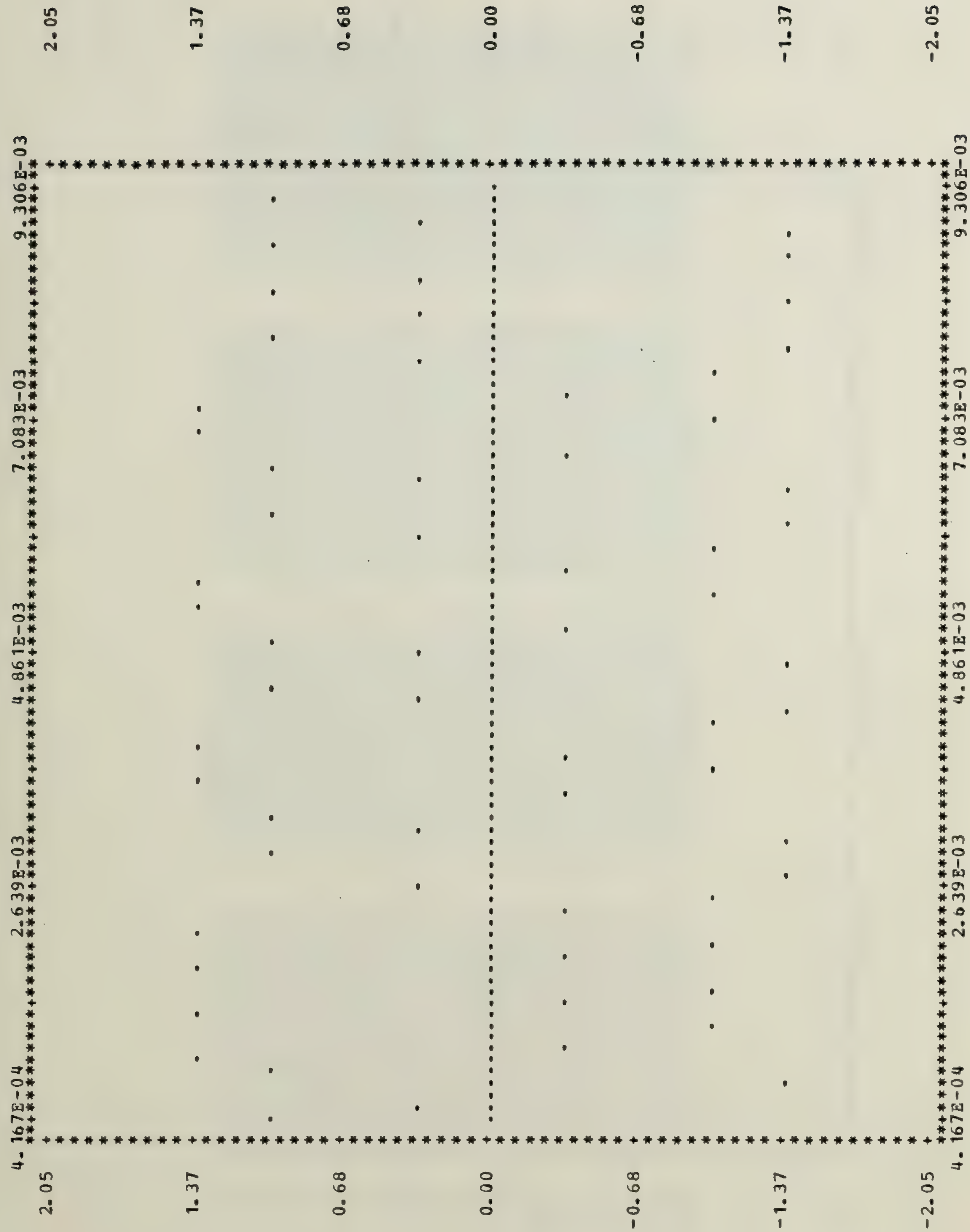

0.000E+00
2.222E-03
4.444E-03
6.667E-03
8.889E-03
2.05
1.37
0.68
0.00
-0.68
-1.37
-2.05


24.95
20.79
16.64
12.48
8.32
4.16
0.00
0.00
59400.00
118800.00
178200.00
237600.00

| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.99999999999999883E+00 | 0.14698904271077763E-14 | 0.21982415887578810E-13 | 0.24424906541753344E-14 |
| 2 | 0.14285950977324444E+01 | 0.43938047543711142E+00 | -0.16994320274213395E+01 | 0.43868836060089 58E+00 |
| 3 | 0.19041231243113341E+01 | 0.82840874959538819E+00 | -0.63968248883996620E+01 | 0.13493183492171100E+01 |
| 4 | 0.25532157885240060E+01 | 0.18793029634204028E+01 | -0.18625446220193388E+02 | 0.10269231662956630E+02 |
| 5 | 0.28952891683411145E+01 | 0.19124220142824412E+01 | -0.30393762386033314E+02 | 0.10845705046518863E+02 |
| 6 | 0.35933196779491141E+01 | 0.31843097312263340E+01 | -0.64026918309457779E+02 | 0.33669258987618804E+02 |
| 7 | 0.39304643411172416E+01 | 0.32830457871474481E+01 | -0.89400379345453391E+02 | 0.22832477281932220E+02 |
| 8 | 0.31763748833367992E+01 | 0.31454364539734428E+01 | -0.40145076890684421E+02 | 0.24708095503393391E+02 |
| 9 | 0.29091093956305561E+01 | 0.21513049992525531E+01 | -0.30954635700674497E+02 | 0.11351746154766698E+02 |
| 10 | 0.23101136641512272E+01 | 0.14489362371016654E+01 | -0.13019851079449930E+02 | 0.50722640667945546E+01 |
| 11 | 0.19296351101333316E+01 | 0.97445621395398873E+00 | -0.67131887575622222E+01 | 0.19686960110138779E+01 |
| 12 | 0.17363337975928557E+01 | 0.68027116027733372E+00 | -0.44833541756800031E+01 | 0.98589415894865534E+00 |
| 13 | 0.26360688597277789E+01 | 0.17943581241550069E+01 | -0.21864589074253396E+02 | 0.72416985640946648E+01 |
| 14 | 0.38034813443652238E+01 | 0.36965405559311117E+01 | -0.79094430267618886E+02 | 0.30199467892593399E+02 |
| 15 | 0.49864744388906447E+01 | 0.66431867068596601E+01 | -0.18208133008381155E+03 | 0.18513169841657733E+03 |
| 16 | 0.66627884012896627E+01 | 0.11979992634857765E+02 | -0.49281164821165519E+03 | 0.30537802662866640E+03 |
| 17 | 0.85029524680167774E+01 | 0.18287726286800051E+02 | -0.10321924119327795E+04 | 0.10402298456203365E+04 |
| 18 | 0.96073745410405777E+01 | 0.22598461106929997E+02 | -0.15461300758967706E+04 | 0.12977252658812222E+04 |
| 19 | 0.10507497805331153E+02 | 0.27601550758456778E+02 | -0.20422636304766643E+04 | 0.18677165406394482E+04 |
| 20 | 0.10676957770340667E+02 | 0.31483954149527713E+02 | -0.21382689862347739E+04 | 0.27450051358579946E+04 |
| 21 | 0.11599989389333332E+02 | 0.34403837978335590E+02 | -0.27792515531935542E+04 | 0.30521096441953352E+04 |
| 22 | 0.12017300384636690E+02 | 0.38981940484898874E+02 | -0.30367606708139772E+04 | 0.41575438440771150E+04 |
| 23 | 0.11504690259682273E+02 | 0.35143215833449966E+02 | -0.26562791304194467E+04 | 0.38872249925263328E+04 |
| 24 | 0.13867577055810111E+02 | 0.32912823902601002E+02 | -0.48045320684816680E+04 | 0.43731776035549112E+04 |
| 25 | 0.11823453377535549E+02 | 0.31940359779053371E+02 | -0.28653922956925525E+04 | 0.39802262719695517E+04 |
| 26 | 0.98986431640089999E+01 | 0.27865653053825517E+02 | -0.16494603734559986E+04 | 0.21434504993970009E+04 |
| 27 | 0.93382360970524430E+01 | 0.20660927747021554E+02 | -0.14125041350190007E+04 | 0.11367534227811173E+04 |
| 28 | 0.80391498289266604E+01 | 0.19181358663676653E+02 | -0.86791921071121875E+03 | 0.97985613348114420E+03 |
| 29 | 0.64843031301271103E+01 | 0.10118336376384492E+02 | -0.45155079755541196E+03 | 0.22415528613476600E+03 |
| 30 | 0.52298418531470119E+01 | 0.67187950906403344E+01 | -0.22452241979610023E+03 | 0.13849752227993374E+03 |
| 31 | 0.36898936916172448E+01 | 0.33261506654947782E+01 | -0.71261532176310072E+02 | 0.26466776319711176E+02 |
| 32 | 0.24450944032305528E+01 | 0.24531815993344463E+01 | -0.14597894164896632E+02 | 0.15585680627972297E+02 |
| 33 | 0.18499484522387445E+01 | 0.13512027515752551E+01 | -0.52227999893714443E+01 | 0.41386537502108896E+01 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.9999999999999883E+02 | 0.9999999999999780E+02 | 0.9999999999999675E+02 | 0.9999999999999573E+02 |
| 2 | 0.1428595097732444E+03 | 0.2475870623947915E+03 | 0.4829849974023537E+03 | 0.1021612251472394E+04 |
| 3 | 0.1904123124311341E+03 | 0.4445809534636610E+03 | 0.1174791159008225E+04 | 0.3354934496273409E+04 |
| 4 | 0.2553215788524060E+03 | 0.8379420796554763E+03 | 0.3252014048362047E+04 | 0.1421333891581056E+05 |
| 5 | 0.2895289168341145E+03 | 0.1027599716245315E+04 | 0.4209678871240456E+04 | 0.1923405871340475E+05 |
| 6 | 0.3593319677949141E+03 | 0.1606441294185059E+04 | 0.8332290249062690E+04 | 0.4869106076961808E+05 |
| 7 | 0.3930464341172416E+03 | 0.1869876526650392E+04 | 0.1001870110919312E+05 | 0.5807186070483972E+05 |
| 8 | 0.3176374883367992E+03 | 0.1320333948912472E+04 | 0.6549247609944333E+04 | 0.3629288523954959E+05 |
| 9 | 0.2909109395630561E+03 | 0.1059270942500601E+04 | 0.4456558088824118E+04 | 0.2069279088431962E+05 |
| 10 | 0.2310113664151272E+03 | 0.6771072016029054E+03 | 0.2323273659349692E+04 | 0.8838397787837077E+04 |
| 11 | 0.1929635110133316E+03 | 0.4688203310073662E+03 | 0.1297939305760345E+04 | 0.3900492110338555E+04 |
| 12 | 0.1736333797592857E+03 | 0.3688323505337797E+03 | 0.8699417987316032E+03 | 0.2195575645600335E+04 |
| 13 | 0.2636068859727789E+03 | 0.8725273576140085E+03 | 0.3323895875929871E+04 | 0.1387981839741076E+05 |
| 14 | 0.3803481344365238E+03 | 0.1812604548730620E+04 | 0.9879885663276686E+04 | 0.5878347959019476E+05 |
| 15 | 0.4986474443889064 7E+03 | 0.3144168216950080E+04 | 0.2385382486185835E+05 | 0.2107014195878238E+06 |
| 16 | 0.6662788401289627E+03 | 0.5625294198886866E+04 | 0.5428072014931859E+05 | 0.5700628197649636E+06 |
| 17 | 0.8502952468016774E+03 | 0.9040504969728506E+04 | 0.1129336708054627E+06 | 0.1591515288375290E+07 |
| 18 | 0.9607374541040577E+03 | 0.1146741220676953E+05 | 0.1574418355567692E+06 | 0.2385295286847671E+07 |
| 19 | 0.1050749780533153E+04 | 0.1377330453799191E+05 | 0.2078610365418646E+06 | 0.3456057121310205E+07 |
| 20 | 0.1067695777034067E+04 | 0.1451665418376697E+05 | 0.2283547269907559E+06 | 0.3996492271439894E+07 |
| 21 | 0.1159998938933332E+04 | 0.1686195534311983E+05 | 0.2819584031049236E+06 | 0.5206323746228511E+07 |
| 22 | 0.1201730038463690E+04 | 0.1830076296146439E+05 | 0.3272181285395482E+06 | 0.6544136345927588E+07 |
| 23 | 0.1150469025968273E+04 | 0.1671496816463538E+05 | 0.2848021016675684E+06 | 0.5476398000266383E+07 |
| 24 | 0.1386757705581011E+04 | 0.2248933890624060E+05 | 0.4167027751216249E+06 | 0.8697347367568203E+07 |
| 25 | 0.1182345337753549E+04 | 0.1714150059520187E+05 | 0.2935171598016949E+06 | 0.5764602626170882E+07 |
| 26 | 0.9898643164009000E+03 | 0.1255701330116690E+05 | 0.1883495169786224E+06 | 0.3169918906461329E+07 |
| 27 | 0.9338236097052430E+03 | 0.1076569718738443E+05 | 0.1429069220727481E+06 | 0.2100176320454131E+07 |
| 28 | 0.8039149828926604E+03 | 0.8361747504897042E+04 | 0.1019465160049566E+06 | 0.1386855086752077E+07 |
| 29 | 0.6484303130127103E+03 | 0.5206334009599722E+04 | 0.4771402657258681E+05 | 0.4769099677938831E+06 |
| 30 | 0.5229841853147019E+03 | 0.3400285294866219E+04 | 0.2542637190179922E+05 | 0.2121683286787419E+06 |
| 31 | 0.3689893691617248E+03 | 0.1690820461427659E+04 | 0.8867626954119426E+04 | 0.5101580138270073E+05 |
| 32 | 0.2445094403230529E+03 | 0.8407136424050373E+03 | 0.3511385920660843E+04 | 0.1646676738681702E+05 |
| 33 | 0.1849948452238745E+03 | 0.4760000000000027E+03 | 0.1434984535671693E+04 | 0.4772000000000989E+04 |

```
MODULATION TECHNIQUE = OQPSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           2
BIT RATE =  0.2400000000000000E+04
CARRIER FREQUENCY =       2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =            0 DEGREES
TIME BETWEEN SAMPLES =  0.1388888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =         64
SEED FOR RANDOM NUMBER GENERATOR =         1
NUMBER OF TIMES SIMULATION REPEATS =       100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.4302799744926702E+03   0.1192673508267923E+05
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.2766466990808353E+05   0.6316161942013623E+08
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.3332128031283272E+05   0.9992700833249297E+08
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.1303878710583849E+02   0.1973876906051148E+03
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.8383233305479857E+03   0.1249052537633781E+07
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.1009735767055537E+04   0.2071291243641945E+07
```

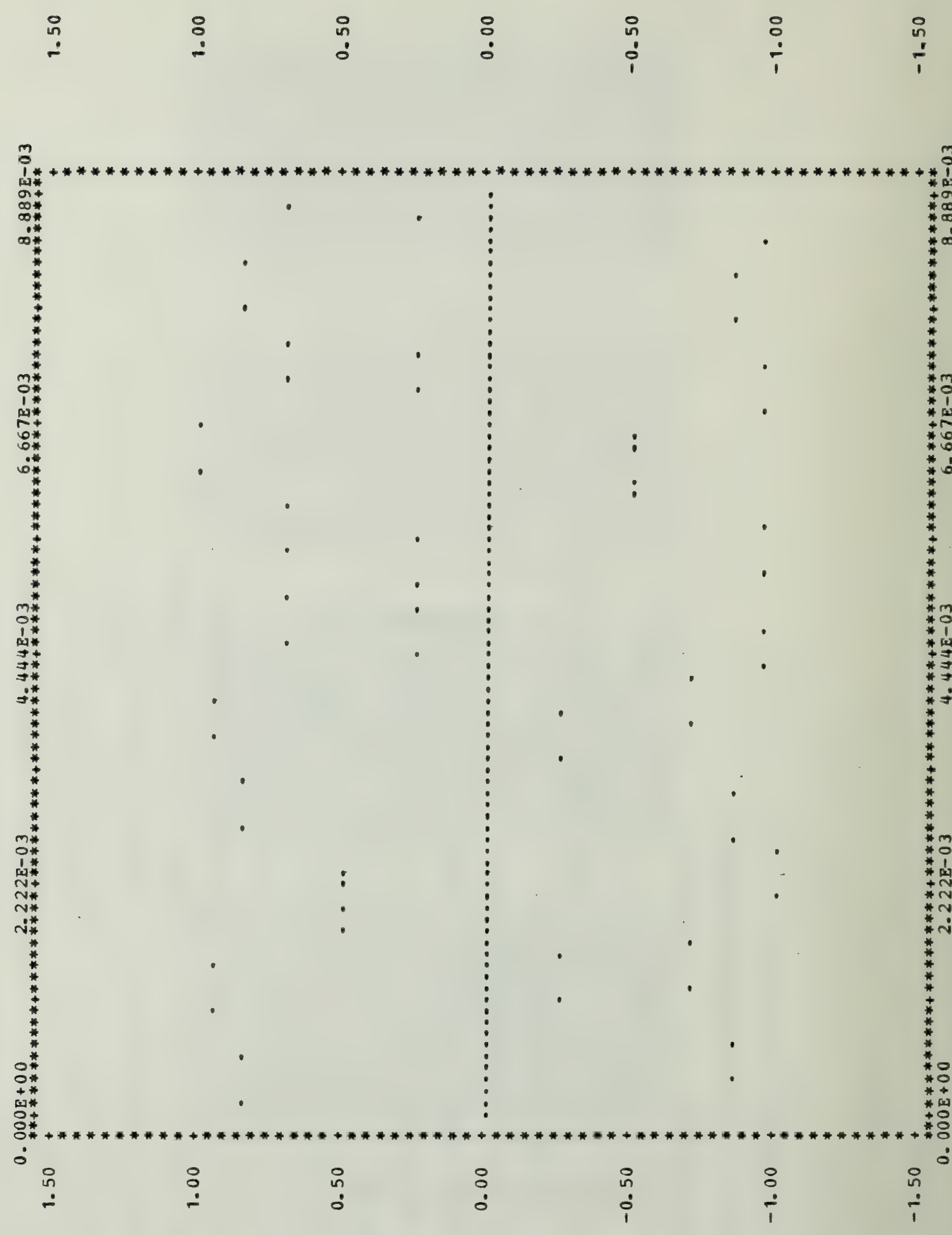

2.05
1.37
0.68
0.00
-0.68
-1.37
-2.05
4.167E-04
2.639E-03
4.861E-03
7.083E-03
9.306E-03

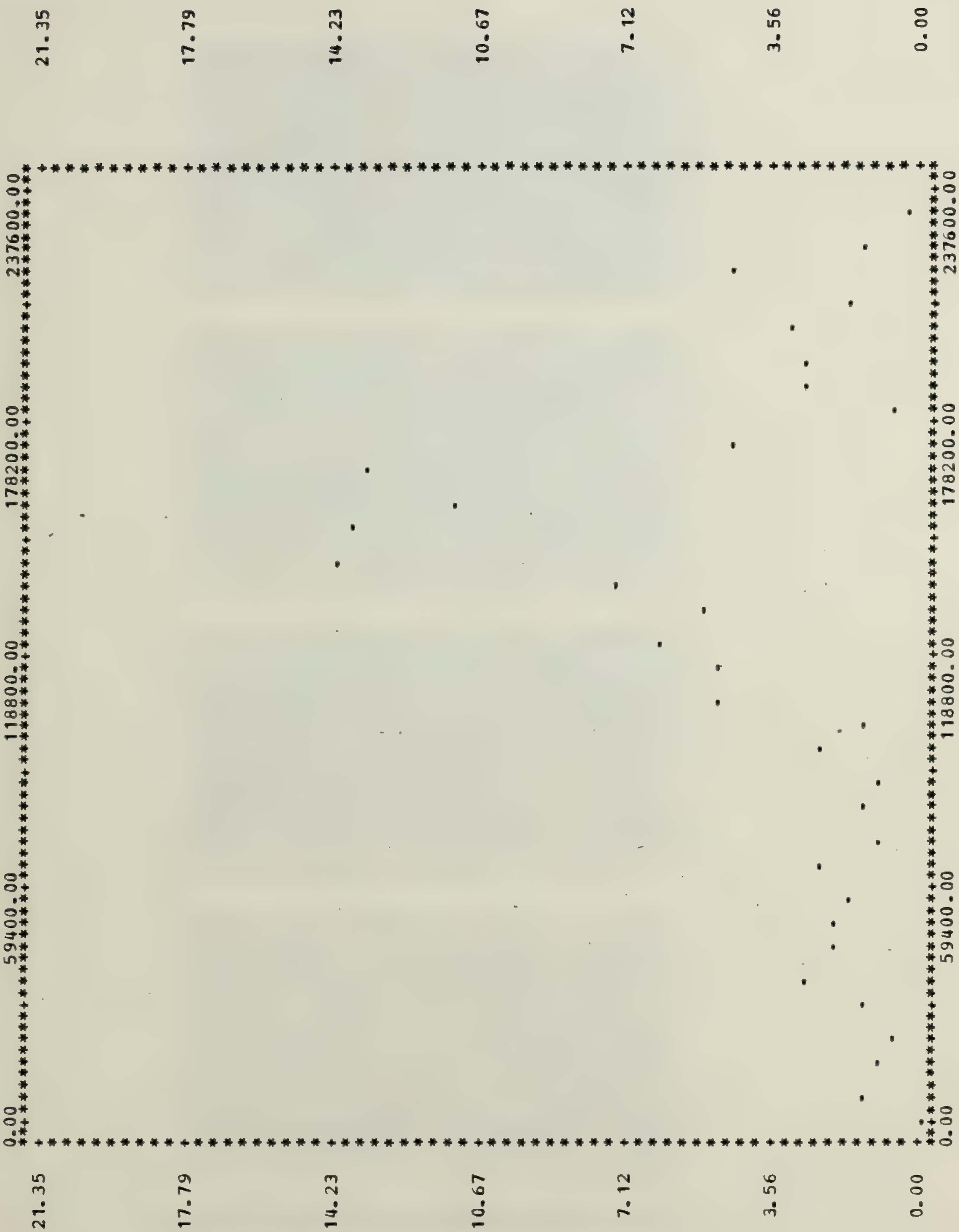

30.57
25.47
20.38
15.28
10.19
5.09
0.00
0.00
59400.00
118800.00
178200.00
237600.00

| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.9999999999999913E+00 | 0.1561758578801998E-14 | 0.1598721155460225E-13 | 0.2664535259100376E-14 |
| 2 | 0.1372323826180889E+01 | 0.4068678643542643E+00 | -0.1386376962675256E+01 | 0.4041815975386704E+00 |
| 3 | 0.2001396615374287E+01 | 0.9269424591664253E+00 | -0.7733658389670000E+01 | 0.1862322872670187E+01 |
| 4 | 0.2772862445069488E+01 | 0.1678068152390965E+01 | -0.2623023671424697E+02 | 0.8004509154649167E+01 |
| 5 | 0.3251532646964970E+01 | 0.2932577358584998E+01 | -0.4366853012949963E+02 | 0.2588931213699834E+02 |
| 6 | 0.3777641084824826E+01 | 0.4020832039397192E+01 | -0.7125587489602146E+02 | 0.7882832999429900E+02 |
| 7 | 0.3829098409125102E+01 | 0.3882607022946678E+01 | -0.7927245839330759E+02 | 0.3918838665086150E+02 |
| 8 | 0.2961659942016187E+01 | 0.3191184577366257E+01 | -0.3117942587408732E+02 | 0.2612505891404616E+02 |
| 9 | 0.2968246427181287E+01 | 0.2173213541787000E+01 | -0.3417545686971072E+02 | 0.1113594383145398E+02 |
| 10 | 0.2148359327285216E+01 | 0.1060831343738275E+01 | -0.1029303883092354E+02 | 0.2522974975096634E+01 |
| 11 | 0.1345271169743835E+01 | 0.4197276401107519E+00 | -0.1125305770333078E+01 | 0.4717607434503110E+00 |
| 12 | 0.1227933909092941E+01 | 0.5054447847122679E+00 | -0.7207547009699091E+00 | 0.3665403064081190E+00 |
| 13 | 0.2439525645946195E+01 | 0.1781420668748019E+01 | -0.1588289360955499E+02 | 0.9490600068457177E+01 |
| 14 | 0.3624460779686187E+01 | 0.2958165706320683E+01 | -0.6557803973417343E+02 | 0.2992394022408189E+02 |
| 15 | 0.5344290929937282E+01 | 0.8707443216668699E+01 | -0.2327526112313629E+03 | 0.2336896609052334E+03 |
| 16 | 0.6817466947394810E+01 | 0.8177726358497095E+01 | -0.5351265563927940E+03 | 0.1713281150266957E+03 |
| 17 | 0.7712888151201321E+01 | 0.2474902892394821E+02 | -0.6993895599099603E+03 | 0.1906132600582511E+04 |
| 18 | 0.9428086215184937E+01 | 0.1931615055428528E+02 | -0.1470236620891471E+04 | 0.8926570202434632E+03 |
| 19 | 0.1072214978861974E+02 | 0.2805072869446679E+02 | -0.2185062282026585E+04 | 0.1838937522132344E+04 |
| 20 | 0.1081933826880679E+02 | 0.3440851415469506E+02 | -0.2210060342212764E+04 | 0.2931776835225231E+04 |
| 21 | 0.1199506255571578E+02 | 0.3580001811338520E+02 | -0.3041862605032027E+04 | 0.3576329272194000E+04 |
| 22 | 0.1128875410898178E+02 | 0.3443719159620903E+02 | -0.2585353070858207E+04 | 0.2812818126869951E+04 |
| 23 | 0.1122041374376493E+02 | 0.3976866880087194E+02 | -0.2372557155779416E+04 | 0.5281536943398933E+04 |
| 24 | 0.1239679867770296E+02 | 0.3228274309630087E+02 | -0.3399416891619934E+04 | 0.3179717696454303E+04 |
| 25 | 0.1194093966023447E+02 | 0.3512070429981194E+02 | -0.3060399165727626E+04 | 0.3009353203186677E+04 |
| 26 | 0.9561857571851002E+01 | 0.2543911177410117E+02 | -0.1477814600152428E+04 | 0.1874123783999903E+04 |
| 27 | 0.1060742458462544E+02 | 0.3139753080593636E+02 | -0.2076204826664873E+04 | 0.2806095691376629E+04 |
| 28 | 0.8728674900290654E+01 | 0.2466868494522429E+02 | -0.1082343896000136E+04 | 0.2016793419812381E+04 |
| 29 | 0.6834317564181064E+01 | 0.1211209931277713E+02 | -0.5263413260618458E+03 | 0.4487766787298260E+03 |
| 30 | 0.5256141872574309E+01 | 0.5747778732450851E+01 | -0.2337931678447483E+03 | 0.7781670547209961E+02 |
| 31 | 0.3639693642400281E+01 | 0.2758452726478166E+01 | -0.6782334107449190E+02 | 0.2464070669964832E+02 |
| 32 | 0.2386927083876014E+01 | 0.1399515226938646E+01 | -0.1523753288228533E+02 | 0.4542469052879014E+01 |
| 33 | 0.1732050807568335E+01 | 0.0000000000000000E+00 | -0.4392304845407262E+01 | 0.3552713678800501E-13 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.99999999999999913E+02 | 0.99999999999999842E+02 | 0.99999999999999767E+02 | 0.99999999999999698E+02 |
| 2 | 0.13723238261808890E+03 | 0.22860718696144770E+03 | 0.42587703546187470E+03 | 0.85901974819430020E+03 |
| 3 | 0.20013966153742870E+03 | 0.49232614466064120E+03 | 0.13815361173789060E+04 | 0.42273252826483000E+04 |
| 4 | 0.27728624450694880E+03 | 0.93500536101437970E+03 | 0.36171468552061920E+04 | 0.15520765672363760E+05 |
| 5 | 0.32515326469649700E+03 | 0.13475716139278170E+04 | 0.66636733666229480E+04 | 0.37307720504661140E+05 |
| 6 | 0.37776410848248260E+03 | 0.18251195884759710E+04 | 0.10704238303682780E+05 | 0.74451678780946950E+05 |
| 7 | 0.38290984091251020E+03 | 0.18505777557948160E+04 | 0.10398485984622110E+05 | 0.64879419893491130E+05 |
| 8 | 0.29616599420161870E+03 | 0.11930702343735920E+04 | 0.57281764536309930E+04 | 0.30763952598809280E+05 |
| 9 | 0.29682464271812870E+03 | 0.10961968258843610E+04 | 0.45817038782901950E+04 | 0.20851336291912660E+05 |
| 10 | 0.21483593272852160E+03 | 0.56656708294342760E+03 | 0.16991755886038370E+04 | 0.55549991351608280E+04 |
| 11 | 0.13452711697438350E+03 | 0.22252848838535900E+03 | 0.42360199855240090E+03 | 0.89284609939590430E+03 |
| 12 | 0.12279339090929410E+03 | 0.20082120219654160E+03 | 0.36614568440878440E+03 | 0.70030732607364430E+03 |
| 13 | 0.24395256459461950E+03 | 0.77148918392897380E+03 | 0.28676565128931590E+04 | 0.12009171933040390E+05 |
| 14 | 0.36244607796861870E+03 | 0.16065299992740880E+04 | 0.82832677591606080E+04 | 0.48226732708316600E+05 |
| 15 | 0.53442909299372820E+03 | 0.37181814328311910E+04 | 0.30625579690124550E+05 | 0.28560402950926630E+06 |
| 16 | 0.68174669473948100E+03 | 0.54573804673732840E+04 | 0.48808306111982670E+05 | 0.47430213676394850E+06 |
| 17 | 0.77128881512013210E+03 | 0.83990182267650460E+04 | 0.11250537977123580E+06 | 0.17253713058286220E+07 |
| 18 | 0.94280862151849370E+03 | 0.10801179872970270E+05 | 0.14070194117891270E+06 | 0.20052078439387800E+07 |
| 19 | 0.10722149788619740E+04 | 0.14273471749712050E+05 | 0.21762777889155630E+06 | 0.36370400113847740E+07 |
| 20 | 0.10819338268806790E+04 | 0.15112250958801510E+05 | 0.24609601504286980E+06 | 0.44402735428828340E+07 |
| 21 | 0.11995062555715780E+04 | 0.17932354364778620E+05 | 0.31233657148403760E+06 | 0.60733807793878690E+07 |
| 22 | 0.11288754108981780E+04 | 0.16152878901329990E+05 | 0.26301513325539040E+06 | 0.46789656798257140E+07 |
| 23 | 0.11220413743764930E+04 | 0.16526866669413210E+05 | 0.29387959326236780E+06 | 0.59888588158549510E+07 |
| 24 | 0.12396798677702960E+04 | 0.18564053312083580E+05 | 0.31972668200304200E+06 | 0.61400296059563070E+07 |
| 25 | 0.11940939660234470E+04 | 0.17735553722617440E+05 | 0.30078040596123070E+06 | 0.55935174409861970E+07 |
| 26 | 0.95618575718510020E+03 | 0.11661384088072450E+05 | 0.16844619716227950E+06 | 0.27406912424177720E+07 |
| 27 | 0.10607424584625440E+04 | 0.14360101181639310E+05 | 0.22684709700529830E+06 | 0.40091445123466290E+07 |
| 28 | 0.87286749002906540E+03 | 0.10061176361073610E+05 | 0.13998987000789620E+06 | 0.22314957047509140E+07 |
| 29 | 0.68343175641810640E+03 | 0.58698874887723160E+04 | 0.58374313573050610E+05 | 0.65013974846355170E+06 |
| 30 | 0.52561418725743100E+03 | 0.33317328329755310E+04 | 0.23631459093499320E+05 | 0.18132342449670170E+06 |
| 31 | 0.36396936424002810E+03 | 0.15978238009742410E+04 | 0.80149593206903990E+04 | 0.44798121972069060E+05 |
| 32 | 0.23869270838760140E+03 | 0.70829409784101140E+03 | 0.24087C162748201 0E+04 | 0.89773114038411600E+04 |
| 33 | 0.17320508075683350E+03 | 0.29999999999998121E+03 | 0.51961524227017480E+03 | 0.89999999999988742E+03 |

```
MOCULATION TECHNIQUE = MPSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =          3
BIT RATE =  0.3600C00000000000E+04
CARRIER FREQUENCY =       2400 HZ
MAXIMUM CARRIER AMPLITUDE =          1 VOLT(S)
INITIAL PHASE ANGLE =          0 DEGREES
TIME BETWEEN SAMPLES =  0.1388888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =         64
SEED FOR RANDOM NUMBER GENERATOR =          1
NUMBER OF TIMES SIMULATION REPEATS =        100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.2100020931141337E+03   0.2899383925937582E+04
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.9600995280888690E+04   0.8204154937490084E+07
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.8272550682771584E+04   0.7152001918873632E+07
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.6363699791337385E+01   0.4884355155646109E+02
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.2909392509360209E+03   0.1690890175696266E+06
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.2506833540233813E+03   0.1586940989808761E+06
```

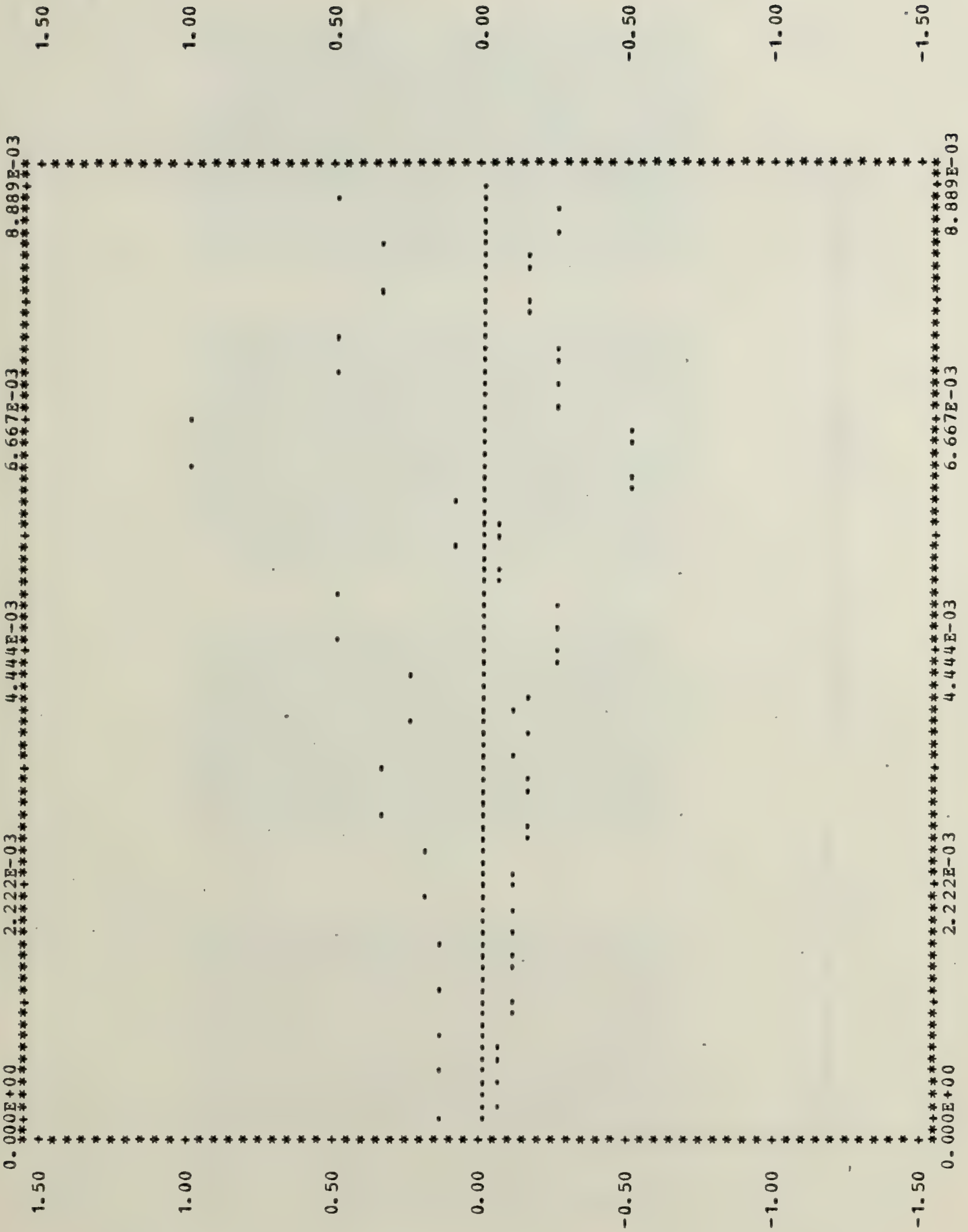

0.000E+00
2.222E-03
4.444E-03
6.667E-03
8.889E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50

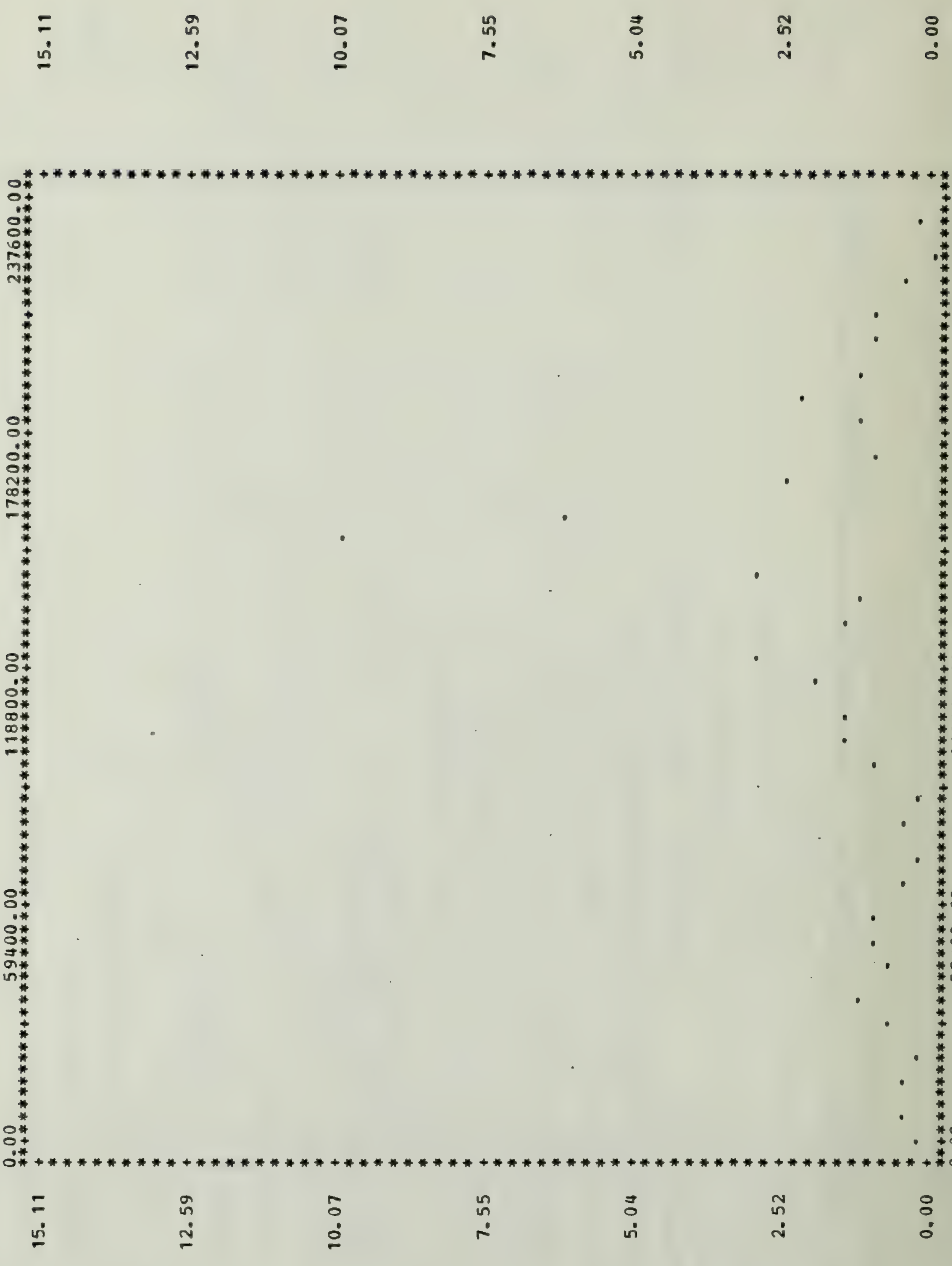

0.00
59400.00
118800.00
178200.00
237600.00
21.35
17.79
14.23
10.67
7.12
3.56
0.00

| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.5852691193458317E+00 | 0.1489495534749113E+00 | 0.2473688383542196E+00 | 0.4064676775736267E-01 |
| 2 | 0.8991336804147069E+00 | 0.2959176116067564E+00 | 0.2789695789784579E+00 | 0.2558696262554949E+00 |
| 3 | 0.1279592076732864E+01 | 0.3929809053583419E+00 | -0.7796903977805032E+00 | 0.5674271316067037E+00 |
| 4 | 0.1737356302159723E+01 | 0.6936742970218630E+00 | -0.4345593469482651E+01 | 0.1075933227133294E+01 |
| 5 | 0.2121975123431636E+01 | 0.1523830949025344E+01 | -0.8949356079797202E+01 | 0.6471426678194621E+01 |
| 6 | 0.2424237800675234E+01 | 0.1477847946943727E+01 | -0.1574379606014421E+02 | 0.6672349398345201E+01 |
| 7 | 0.2493407977880165E+01 | 0.1246925063550747E+01 | -0.1813201191600308E+02 | 0.4341297777074292E+01 |
| 8 | 0.2478804546055675E+01 | 0.2098145396937312E+01 | -0.1694930495176807E+02 | 0.9359754143637726E+01 |
| 9 | 0.2071593329231943E+01 | 0.1038070914171632E+01 | -0.8959518397406967E+01 | 0.2630686655844620E+01 |
| 10 | 0.1792265854082727E+01 | 0.8172058974659458E+00 | -0.4858409033532932E+01 | 0.1708412203826555E+01 |
| 11 | 0.1338421965915424E+01 | 0.4602410319466023E+00 | -0.1076573354148309E+01 | 0.4930109231783348E+00 |
| 12 | 0.1206794109459943E+01 | 0.3465146025707462E+00 | -0.5824924961494571E+00 | 0.2784383478326526E+00 |
| 13 | 0.1701079214725041E+01 | 0.8240034102520097E+00 | -0.3676238752314777E+01 | 0.1845216445410600E+01 |
| 14 | 0.2834150610551414E+01 | 0.1823855047315213E+01 | -0.2841376075861841E+02 | 0.1025688233426690E+02 |
| 15 | 0.3743807553618862E+01 | 0.3367091128248024E+01 | -0.7536218978763767E+02 | 0.2764443212163627E+02 |
| 16 | 0.4695195021029540E+01 | 0.5878572281542108E+01 | -0.1604268845957975E+03 | 0.7733863012447296E+02 |
| 17 | 0.5610648269528947E+01 | 0.7832247511634661E+01 | -0.2870389196847128E+03 | 0.1576959426040605E+03 |
| 18 | 0.6706908951556747E+01 | 0.1336843509358619E+02 | -0.4883380690179577E+03 | 0.4717187808135241E+03 |
| 19 | 0.7233528197904389E+01 | 0.1179079685993331E+02 | -0.6391723454185712E+03 | 0.3894371981662816E+03 |
| 20 | 0.7838475981757254E+01 | 0.1550564901810932E+02 | -0.8148308730885305E+03 | 0.8369144820736628E+03 |
| 21 | 0.8345623481532100E+01 | 0.2143264337995021E+02 | -0.9600418962579780E+03 | 0.1423931246664269E+04 |
| 22 | 0.8944133416090430E+01 | 0.1849543486323111E+02 | -0.1217075995286470E+04 | 0.1263290690873313E+04 |
| 23 | 0.8814783978046604E+01 | 0.2037175020305552E+02 | -0.1196920188977485E+04 | 0.9463713933138024E+03 |
| 24 | 0.8749509753732253E+01 | 0.1838659826724127E+02 | -0.1168060381440785E+04 | 0.8402328493104333E+03 |
| 25 | 0.8285982175052635E+01 | 0.1666403923810872E+02 | -0.9723427666802374E+03 | 0.6372386843278655E+03 |
| 26 | 0.7248418801596032E+01 | 0.1376932655732141E+02 | -0.6340385567820426E+03 | 0.5186332515460535E+03 |
| 27 | 0.6082018000723616E+01 | 0.8179763542192111E+01 | -0.3674332594430192E+03 | 0.1611004446780380E+03 |
| 28 | 0.5574426942061457E+01 | 0.8531145086994827E+01 | -0.2667689307470644E+03 | 0.3277737781942465E+03 |
| 29 | 0.4540960061879976E+01 | 0.6292988896313568E+01 | -0.1361945600435881E+03 | 0.1056182502126908E+03 |
| 30 | 0.3600823367347476E+01 | 0.2838980150814418E+01 | -0.6510362543227304E+02 | 0.2228552737517583E+02 |
| 31 | 0.2979632931111714E+01 | 0.2494350499185918E+01 | -0.3337309524735176E+02 | 0.1665843666245958E+02 |
| 32 | 0.1788877268851379E+01 | 0.8979917766910076E+00 | -0.4784211837438346E+01 | 0.1743915793219237E+01 |
| 33 | 0.1438892582981557E+01 | 0.7161261323388549E+00 | -0.1748123869941282E+01 | 0.9253962560184192E+00 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
| --- | --- | --- | --- | --- |
| 1 | 0.5852691193458317E+02 | 0.4900000000000076E+02 | 0.4226345596729147E+02 | 0.3750000000000090E+02 |
| 2 | 0.8991336804147069E+02 | 0.1101399810746785E+03 | 0.1633003824800721E+03 | 0.2747253888189251E+03 |
| 3 | 0.1279592076732864E+03 | 0.2026406979142283E+03 | 0.3724520780784378E+03 | 0.7766002785739216E+03 |
| 4 | 0.1737356302159724E+03 | 0.3705144474705753E+03 | 0.8929060999836082E+03 | 0.2335831349797537E+04 |
| 5 | 0.2121975123431636E+03 | 0.6011371063997799E+03 | 0.2031302663783766E+04 | 0.7730418455554565E+04 |
| 6 | 0.2424237800675234E+03 | 0.7339998381696989E+03 | 0.2588405071282505E+04 | 0.1024637141588694E+05 |
| 7 | 0.2493407977880165E+03 | 0.7451539157071692E+03 | 0.2517300294087098E+04 | 0.9340289925095110E+04 |
| 8 | 0.2478804546055675E+03 | 0.8221635920514223E+03 | 0.3190123658049523E+04 | 0.1358254136951836E+05 |
| 9 | 0.2071593329231943E+03 | 0.5319189126748201E+03 | 0.1551507389484043E+04 | 0.4948150356240742E+04 |
| 10 | 0.1792265854082727E+03 | 0.4021250730202174E+03 | 0.1033860830638506E+04 | 0.2927878573892384E+04 |
| 11 | 0.1338421965915424E+03 | 0.2247011980472043E+03 | 0.4363030505178581E+03 | 0.9326945526938799E+03 |
| 12 | 0.1206794109459943E+03 | 0.1799401479172257E+03 | 0.3019324775451872E+03 | 0.5492763501245419E+03 |
| 13 | 0.1701079214725041E+03 | 0.3709433870919051E+03 | 0.9466542826798441E+03 | 0.2697528412673160E+04 |
| 14 | 0.2834150610551415E+03 | 0.9838026180131017E+03 | 0.3916876358992104E+04 | 0.1737180587441166E+05 |
| 15 | 0.3743807553618863E+03 | 0.1734951521549920E+04 | 0.9146535856152885E+04 | 0.5276772696201834E+05 |
| 16 | 0.4695195021029540E+03 | 0.2786464284422727E+04 | 0.1879731998681482E+05 | 0.1379916742814523E+06 |
| 17 | 0.5610648269528947E+03 | 0.3923329904088649E+04 | 0.3103750564215353E+05 | 0.2685931834990769E+06 |
| 18 | 0.6706908951556748E+03 | 0.5821737842712236E+04 | 0.5930726451401991E+05 | 0.6740142140150419E+06 |
| 19 | 0.7233528197904389E+03 | 0.6399681908121190E+04 | 0.6449481804020687E+05 | 0.7172431042336427E+06 |
| 20 | 0.7838475981757255E+03 | 0.7679229824451359E+04 | 0.8680894715989773E+05 | 0.1107065923850587E+07 |
| 21 | 0.8345623481532100E+03 | 0.9086774824165068E+04 | 0.1175703281469423E+06 | 0.1725164680980439E+07 |
| 22 | 0.8944133416090430E+03 | 0.9830800307942427E+04 | 0.1260768645819277E+06 | 0.1838165327263264E+07 |
| 23 | 0.8814783978046604E+03 | 0.9786844928065208E+04 | 0.1235746693889336E+06 | 0.1700328056148307E+07 |
| 24 | 0.8749509753732253E+03 | 0.9475665321522469E+04 | 0.1166054561310802E+06 | 0.1570740428182183E+07 |
| 25 | 0.8285982175052636E+03 | 0.8515489945101765E+04 | 0.1007118169017778E+06 | 0.1307951327542595E+07 |
| 26 | 0.7248418801596032E+03 | 0.6617120841507905E+04 | 0.6997921865718614E+05 | 0.6229758430242574E+06 |
| 27 | 0.6082018000723617E+03 | 0.4508890886789629E+04 | 0.3812795207373243E+05 | 0.3534598125194023E+06 |
| 28 | 0.5574426942061457E+03 | 0.3952006936850553E+04 | 0.3319878161079791E+05 | 0.3258829753140927E+06 |
| 29 | 0.4540960061879976E+03 | 0.2685037729093943E+04 | 0.1883442761629288E+05 | 0.1480286217678719E+06 |
| 30 | 0.3600823367347476E+03 | 0.1577651927214189E+04 | 0.7938989447366468E+04 | 0.4427640492853110E+05 |
| 31 | 0.2979632931111714E+03 | 0.1134761939835944E+04 | 0.5030570129153916E+04 | 0.2482170481050692E+05 |
| 32 | 0.1788877268851379E+03 | 0.4089093741937266E+03 | 0.1076028493199732E+04 | 0.3094819564294680E+04 |
| 33 | 0.1438892582981557E+03 | 0.2779376736374803E+03 | 0.6108726113183613E+03 | 0.1441768747581861E+04 |

```
MODULATION TECHNIQUE = MASK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =            3
BIT RATE =  0.3600000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =          0 DEGREES
TIME BETWEEN SAMPLES =  0.1388888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =          1
NUMBER OF TIMES SIMULATION REPEATS =        100
   SUM OF VARIANCES        SUM OF VARIANCES**2
 0.2672011399771625E+02   0.6236464612091026E+02
   SUM OF SKEWNESS          SUM OF SKEWNESS**2
-0.1657760855944905E+04   0.1880354503890174E+07
   SUM OF KURTOSIS          SUM OF KURTOSIS**2
 0.1906467874763258E+03   0.7184370776848387E+04
    MEAN VARIANCE        VARIANCE OF THE VARIANCES
 0.8097004241732196E+00   0.1272792452593831E+01
    MEAN SKEWNESS        VARIANCE OF THE SKEWNESS
-0.5023517745287592E+02   0.5615864353491720E+05
    MEAN KURTOSIS        VARIANCE OF THE KURTOSIS
 0.5777175378070478E+01   0.1900928390728725E+03
```

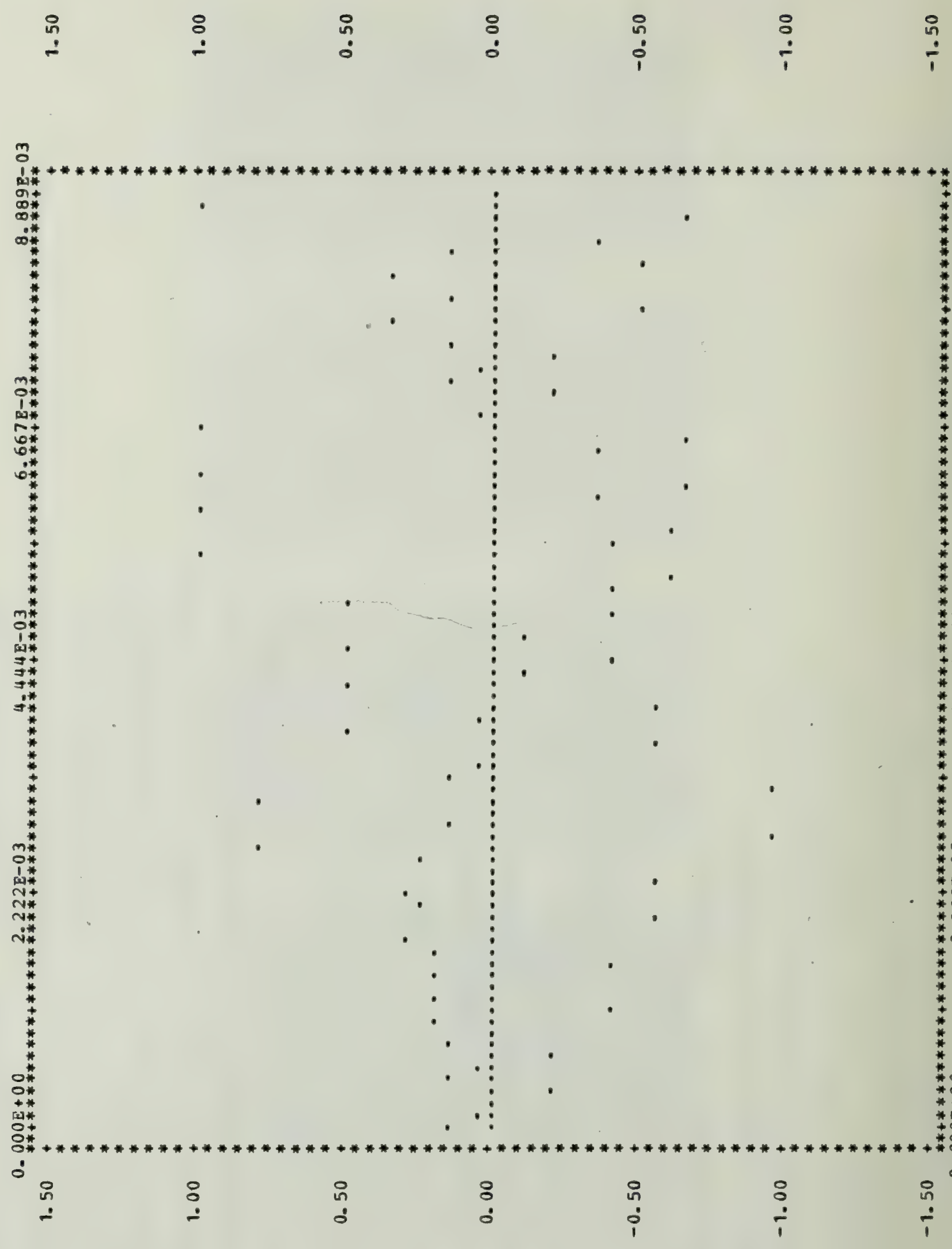

1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50
0.000E+00
2.222E-03
4.444E-03
6.667E-03
8.889E-03


0.00
59400.00
118800.00
178200.00
237600.00
15.11
12.59
10.07
7.55
5.04
2.52
0.00

| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.339357142857 0985E+00 | 0.8384464886964783E-01 | 0.1893382723578415E+00 | 0.2742736031887471E-01 |
| 2 | 0.4004012610889417E+00 | 0.7668190114822222E-01 | 0.2215258893174914E+00 | 0.2891499870294614E-01 |
| 3 | 0.5679613369866736E+00 | 0.7784761505069471E-01 | 0.2883442731501778E+00 | 0.1556475446215472E-01 |
| 4 | 0.6569355772557277E+00 | 0.1457017233934383E+00 | 0.3309802432325256E+00 | 0.6104856817896656E-01 |
| 5 | 0.7161005021434623E+00 | 0.1586238997580487E+00 | 0.3483330687474483E+00 | 0.1090588968757279E+00 |
| 6 | 0.8413504533675261E+00 | 0.2506368816797936E+00 | 0.3341500081150626E+00 | 0.2119929827562888E+00 |
| 7 | 0.8750519414575849E+00 | 0.1582141403087646E+00 | 0.2153114973989920E+00 | 0.8553177882653351E-01 |
| 8 | 0.8810672568079662E+00 | 0.2475548419903926E+00 | 0.2967083292659543E+00 | 0.2369340344206032E+00 |
| 9 | 0.7193220063807960E+00 | 0.1455112103205957E+00 | 0.3180583366119436E+00 | 0.5536073992160295E-01 |
| 10 | 0.6400695740593127E+00 | 0.1857408752897842E+00 | 0.4025441093271451E+00 | 0.1852465158214689E+00 |
| 11 | 0.5096201315245369E+00 | 0.1626402382072894E+00 | 0.3366001013572022E+00 | 0.9761198503664166E-01 |
| 12 | 0.4682289421728521E+00 | 0.1574634369096420E+00 | 0.3172125727703493E+00 | 0.9859166537393366E-01 |
| 13 | 0.6648869535479373E+00 | 0.1580594153239595E+00 | 0.3627225421498026E+00 | 0.1048537336552618E+00 |
| 14 | 0.8376250912444208E+00 | 0.2646497831913186E+00 | 0.3375016241785487E+00 | 0.2258943155139170E+00 |
| 15 | 0.1210125608088872E+01 | 0.4229914530943488E+00 | -0.5096043285692129E+00 | 0.4625565690160802E+00 |
| 16 | 0.1517580038320735E+01 | 0.7062878859778058E+00 | -0.1866015564593158E+01 | 0.1681719571938255E+01 |
| 17 | 0.1809198427029523E+01 | 0.1114007724685303E+01 | -0.4305283960734115E+01 | 0.4061604744465562E+01 |
| 18 | 0.2184365983429202E+01 | 0.1357504208894486E+01 | -0.9744491483583519E+01 | 0.7477973881021072E+01 |
| 19 | 0.2187798064034754E+01 | 0.1242471878317058E+01 | -0.1056072077463751E+02 | 0.4506869998249794E+01 |
| 20 | 0.2754817782881566E+01 | 0.2304339434717248E+01 | -0.2331072874845357E+02 | 0.1840676577341128E+02 |
| 21 | 0.3284482283145103E+01 | 0.3566553684367836E+01 | -0.4341407481522182E+02 | 0.4400306198337802E+02 |
| 22 | 0.9147694673424342E+01 | 0.4931457617733018E+01 | -0.1359555809056429E+04 | 0.6507944926457276E+02 |
| 23 | 0.4770282409732731E+01 | 0.2378616824097682E+01 | -0.1692516756760991E+03 | 0.1754223705726594E+02 |
| 24 | 0.2682785373968661E+01 | 0.2279330804925413E+01 | -0.2224917161374269E+02 | 0.1520216675006741E+02 |
| 25 | 0.2264338741105986E+01 | 0.1317600360027486E+01 | -0.1196687807153250E+02 | 0.5562161697791915E+01 |
| 26 | 0.1730042306558164E+01 | 0.8497906580179376E+00 | -0.3919560952908456E+01 | 0.1879382110769463E+01 |
| 27 | 0.1520548002468391E+01 | 0.7665104475471458E+00 | -0.1933518608175382E+01 | 0.1972853904093768E+01 |
| 28 | 0.1268184557381321E+01 | 0.5151865914419896E+00 | -0.6604973057600012E+00 | 0.7869755954404178E+00 |
| 29 | 0.9374132388714257E+00 | 0.2806502823386208E+00 | 0.2713984344570348E+00 | 0.3435098070765137E+00 |
| 30 | 0.7199988342830723E+00 | 0.1763086287440097E+00 | 0.3250195433939274E+00 | 0.7846116889130217E-01 |
| 31 | 0.5315997481872489E+00 | 0.8431479110743091E-01 | 0.2707480644469704E+00 | 0.1669316880037137E-01 |
| 32 | 0.3681907329598634E+00 | 0.6396253104650891E-01 | 0.1920543954296889E+00 | 0.1778817793243678E-01 |
| 33 | 0.2603333333334280E+00 | 0.8905757919335087E-01 | 0.1286237098262513E+00 | 0.2052392227851867E-01 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.3393571428570985E+02 | 0.1981694727890845E+02 | 0.1607624098031789E+02 | 0.1485075176620011E+02 |
| 2 | 0.4004012610689417E+02 | 0.2362362520183549E+02 | 0.1846514292234847E+02 | 0.1745202579792261E+02 |
| 3 | 0.5679613369866736E+02 | 0.3996492192118776E+02 | 0.3241400271345726E+02 | 0.2906212284536081E+02 |
| 4 | 0.6569355772557277E+02 | 0.5758090588238202E+02 | 0.6026599072464039E+02 | 0.7124348934965693E+02 |
| 5 | 0.7161005021434623E+02 | 0.6698375899305872E+02 | 0.7617463139187906E+02 | 0.1018941858527745E+03 |
| 6 | 0.8413504533675261E+02 | 0.9560010982447375E+02 | 0.1331404709635767E+03 | 0.2135596761732665E+03 |
| 7 | 0.8750519414575849E+02 | 0.9223478991543655E+02 | 0.1105186656464672E+03 | 0.1475338504496394E+03 |
| 8 | 0.8810672568079662E+02 | 0.1021358804589603E+03 | 0.1443806707757001E+03 | 0.2375967453151129E+03 |
| 9 | 0.7193220063807960E+02 | 0.6614802470810837E+02 | 0.7106619344194451E+02 | 0.8497279404873148E+02 |
| 10 | 0.6400695740593127E+02 | 0.5935725261733565E+02 | 0.7229491320574764E+02 | 0.1080651844225191E+03 |
| 11 | 0.5096201315245369E+02 | 0.4207265142803029E+02 | 0.4604068490771932E+02 | 0.5828863474887966E+02 |
| 12 | 0.4682289421728521E+02 | 0.3751271448288537E+02 | 0.4056720468043424E+02 | 0.5091240706613379E+02 |
| 13 | 0.6648869535479373E+02 | 0.5985534821689769E+02 | 0.6724844240378106E+02 | 0.8920162650581295E+02 |
| 14 | 0.8376250912444208E+02 | 0.9636190788416297E+02 | 0.1355724593732854E+03 | 0.2188502038919660E+03 |
| 15 | 0.1210125608088872E+03 | 0.1883165525915868E+03 | 0.3398188178268685E+03 | 0.6798641196073320E+03 |
| 16 | 0.1517580038320735E+03 | 0.3002274179827592E+03 | 0.7196460184604044E+03 | 0.1979233294199776E+04 |
| 17 | 0.1809198427029523E+03 | 0.4376066595804551E+03 | 0.1289983654765179E+04 | 0.4361413529881382E+04 |
| 18 | 0.2184365983429202E+03 | 0.6115383916368165E+03 | 0.2078730882486524E+04 | 0.8233101049976278E+04 |
| 19 | 0.2187798064034754E+03 | 0.6016507528528105E+03 | 0.1935506905686699E+04 | 0.6983075996726611E+04 |
| 20 | 0.2754817782881566E+03 | 0.9870317057250584E+03 | 0.4308400407376489E+04 | 0.2165045465334777E+05 |
| 21 | 0.3284482283145103E+03 | 0.1431871201581823E+04 | 0.7609894524842568E+04 | 0.4661105836764793E+05 |
| 22 | 0.9147694673424342E+03 | 0.8856246087975177E+04 | 0.9035106102323497E+05 | 0.9666813766485890E+06 |
| 23 | 0.4770282409732731E+03 | 0.2511042492446221E+04 | 0.1445885907409191E+05 | 0.9014912438871981E+05 |
| 24 | 0.2682785373968661E+03 | 0.9453874859656329E+03 | 0.3944430526314505E+04 | 0.1856332225736484E+05 |
| 25 | 0.2264338741105986E+03 | 0.6431654290900655E+03 | 0.2146899400038132E+04 | 0.8102049904027280E+04 |
| 26 | 0.1730042306558164E+03 | 0.3834339133918852E+03 | 0.9995053040242969E+03 | 0.2906369127358244E+04 |
| 27 | 0.1520548002468391E+03 | 0.3070911570882289E+03 | 0.7450754300789790E+03 | 0.2072581731545704E+04 |
| 28 | 0.1268184557381321E+03 | 0.2118326797108026E+03 | 0.4182206545817384E+03 | 0.9320695629873251E+03 |
| 29 | 0.9374132388714257E+02 | 0.1156587359926651E+03 | 0.1766512182953982E+03 | 0.3185825968811039E+03 |
| 30 | 0.7199988342830723E+02 | 0.6929438638255527E+02 | 0.7849792231899791E+02 | 0.9900825153580844E+02 |
| 31 | 0.5315997481872489E+02 | 0.3660699354691031E+02 | 0.2893595364443705E+02 | 0.2508679483286366E+02 |
| 32 | 0.3681907329598634E+02 | 0.1988873215735653E+02 | 0.1406109698744312E+02 | 0.1182356732979405E+02 |
| 33 | 0.2603333333334280E+02 | 0.1559404478459111E+02 | 0.1148663107049648E+02 | 0.9050603213587212E+01 |

```
MODULATION TECHNIQUE = QASK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           3
BIT RATE =  0.360000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.1388888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =          1
NUMBER OF TIMES SIMULATION REPEATS =        100
   SUM OF VARIANCES           SUM OF VARIANCES**2
 0.4885105807865026E+02   0.1949104911522201E+03
   SUM OF SKEWNESS             SUM OF SKEWNESS**2
-0.5438333348597472E+04   0.1951490781083542E+08
   SUM OF KURTOSIS             SUM OF KURTOSIS**2
 0.6866043785868234E+03   0.9002560164061202E+05
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.1480335093292432E+01   0.3831079860435228E+01
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.1647979802605294E+03   0.5818337955938460E+06
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.2080619329050980E+02   0.2366874319550756E+04
```

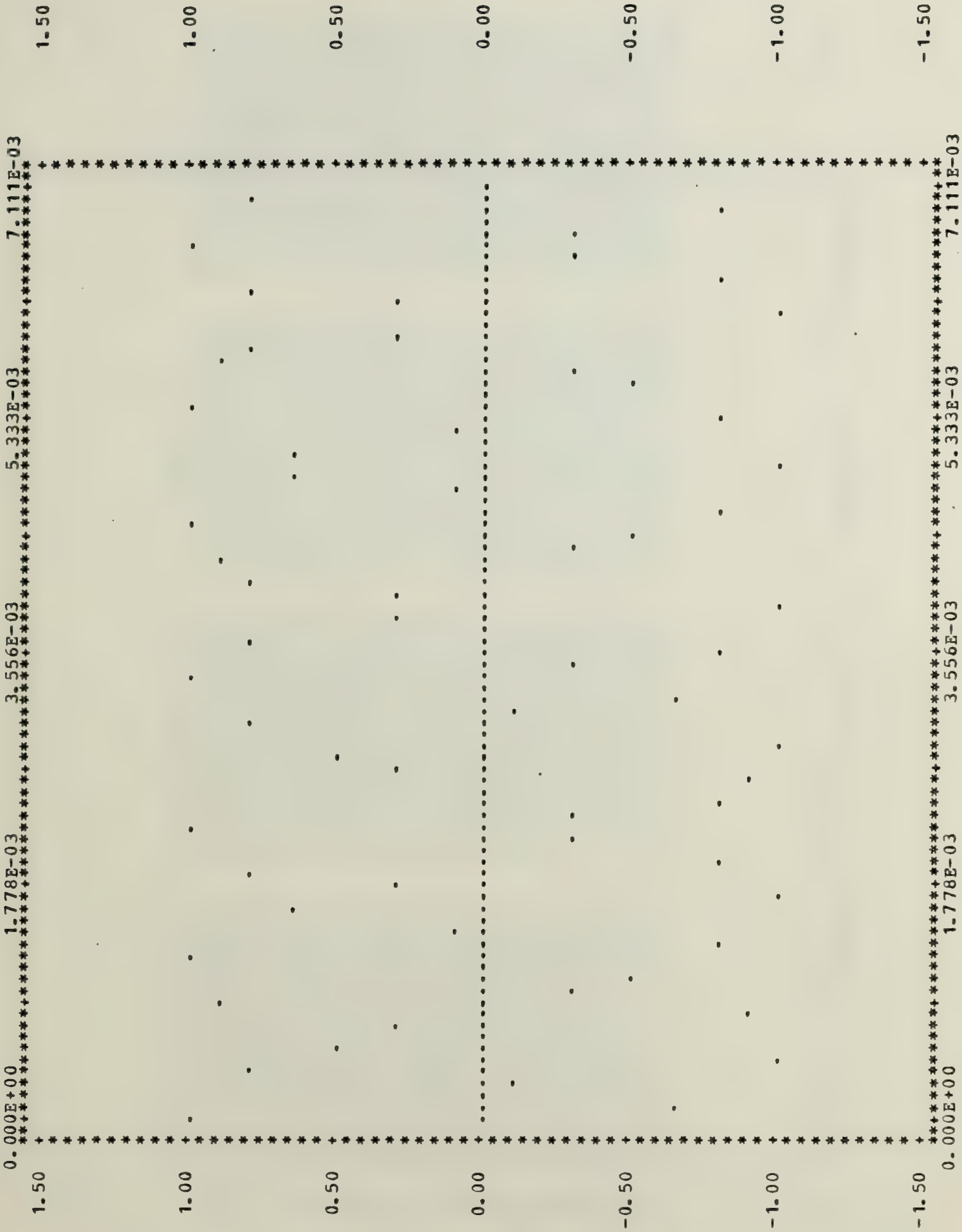

0.000E+00
2.222E-03
4.444E-03
6.667E-03
8.889E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50


19.50
16.25
13.00
9.75
6.50
3.25
0.00
0.00
59400.00
118800.00
178200.00
237600.00

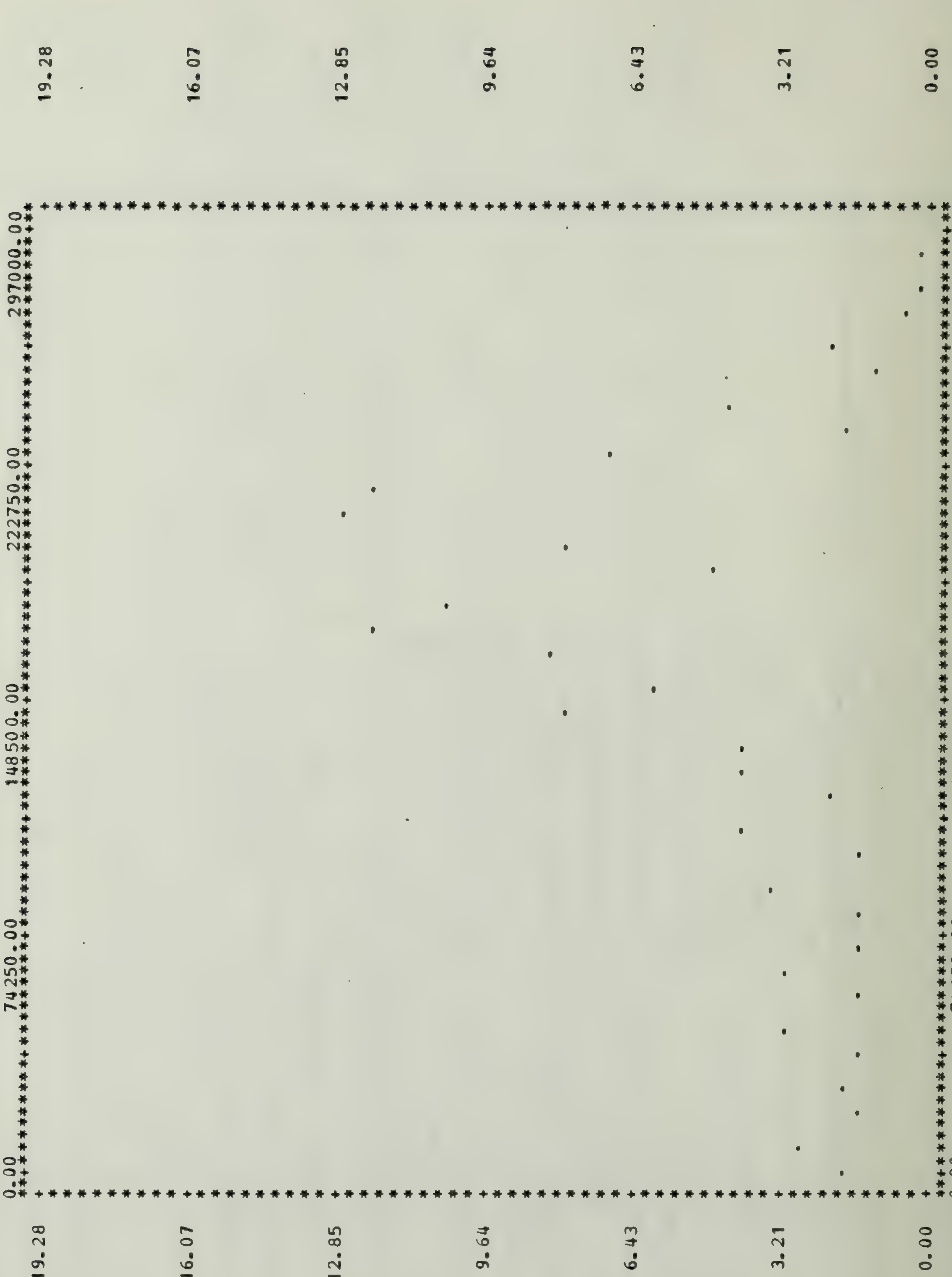
| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.3631071428570145E+00 | 0.9100003707711814E-01 | 0.2048764448173950E+00 | 0.2736509346421254E-01 |
| 2 | 0.4348756578345995E+00 | 0.1077402960376041E+00 | 0.2519993671461299E+00 | 0.3849748104134676E-01 |
| 3 | 0.6750899268356537E+00 | 0.1044729899019833E+00 | 0.3121394810475125E+00 | 0.3187668066555382E-01 |
| 4 | 0.7480786258634735E+00 | 0.1398669466983451E+00 | 0.2881211010787542E+00 | 0.4528694704505194E-01 |
| 5 | 0.9221266466053553E+00 | 0.2237062295889853E+00 | 0.1777804823643814E+00 | 0.1259198848576282E+00 |
| 6 | 0.9941197242066054E+00 | 0.2448428085097390E+00 | 0.8874388910038866E-01 | 0.1407886371080798E+00 |
| 7 | 0.1026103523087546E+01 | 0.2821921014461165E+00 | -0.1299944649149154E-01 | 0.1698637724238030E+00 |
| 8 | 0.1020281417107785E+01 | 0.2329802191693231E+00 | 0.1575997299298781E-01 | 0.1468781252187603E+00 |
| 9 | 0.9039603910129196E+00 | 0.2573160390871251E+00 | 0.2761238684816987E+00 | 0.2538951036403536E+00 |
| 10 | 0.8095873974206875E+00 | 0.2572144259466262E+00 | 0.3532647086406504E+00 | 0.1797884544472488E+00 |
| 11 | 0.6921123568145022E+00 | 0.2192213128955997E+00 | 0.3862627460750074E+00 | 0.1448063331992847E+00 |
| 12 | 0.6486261053527244E+00 | 0.2132730834765532E+00 | 0.3832501053715590E+00 | 0.1314955252811468E+00 |
| 13 | 0.8754097356965494E+00 | 0.2273581893691260E+00 | 0.2541572723549317E+00 | 0.1219793684327222E+00 |
| 14 | 0.1209721959332601E+01 | 0.3494619809652673E+00 | -0.4550529376707417E+00 | 0.4175623429098005E+00 |
| 15 | 0.1588109522148007E+01 | 0.6756164239192016E+00 | -0.2696042172542727E+01 | 0.1254194360641240E+01 |
| 16 | 0.1906974609084911E+01 | 0.1093711640060120E+01 | -0.5825711624576704E+01 | 0.3956890453964988E+01 |
| 17 | 0.2357465080233630E+01 | 0.1554581785324349E+01 | -0.1372480765948216E+02 | 0.6722300394752644E+01 |
| 18 | 0.2752879991704121E+01 | 0.1810579104882114E+01 | -0.2526590473117368E+02 | 0.8308910926769993E+01 |
| 19 | 0.3101276851917939E+01 | 0.2283995368334145E+01 | -0.3911214697839433E+02 | 0.1311113970217036E+02 |
| 20 | 0.3698634891445204E+01 | 0.4165581587386805E+01 | -0.6713629916099845E+02 | 0.6390920635610388E+02 |
| 21 | 0.4176967218993768E+01 | 0.4947265505598236E+01 | -0.1025814110206936E+03 | 0.8320053858876270E+02 |
| 22 | 0.1333645961824457E+02 | 0.7208745191674609E+01 | -0.4376844150885777E+04 | 0.1722999930817896E+03 |
| 23 | 0.6989573530171478E+01 | 0.7325599487433367E+01 | -0.5710800411367075E+03 | 0.2122360733839823E+03 |
| 24 | 0.4042855384246694E+01 | 0.3686258733068045E+01 | -0.9397003552612041E+02 | 0.5505842656900205E+02 |
| 25 | 0.3311545045919114E+01 | 0.2169770846806709E+01 | -0.4870033513540695E+02 | 0.1638713147647758E+02 |
| 26 | 0.3373086709604516E+01 | 0.2558059327713910E+01 | -0.5178306112892427E+02 | 0.1990842663176574E+02 |
| 27 | 0.2677625104741695E+01 | 0.1830946288086045E+01 | -0.2260532808915164E+02 | 0.9890794785552860E+01 |
| 28 | 0.2327844363369372E+01 | 0.1989531364017429E+01 | -0.1176371687561331E+02 | 0.1326199700393431E+02 |
| 29 | 0.1969748167184020E+01 | 0.1246866793002916E+01 | -0.6838808541389447E+01 | 0.3589504978037020E+01 |
| 30 | 0.1366760730506631E+01 | 0.4628643448965760E+00 | -0.1232708015553322E+01 | 0.6635394256326872E+00 |
| 31 | 0.1184996267907273E+01 | 0.3774261054142214E+00 | -0.3882333969399698E+00 | 0.5102456005621498E+00 |
| 32 | 0.8700976947091932E+00 | 0.2753976666023784E+00 | 0.2768835890762174E+00 | 0.1831824042147203E+00 |
| 33 | 0.6506228822272112E+00 | 0.2376138542595925E+00 | 0.4140828375833057E+00 | 0.1758787129718080E+00 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.3631071428570145E+02 | 0.2219368339001313E+02 | 0.1829433993864518E+02 | 0.1696580831036788E+02 |
| 2 | 0.4348756578345995E+02 | 0.2957797308543037E+02 | 0.2596479066801969E+02 | 0.2618296686591356E+02 |
| 3 | 0.6750899268356537E+02 | 0.5591746693179316E+02 | 0.5331262222121557E+02 | 0.5655725325363117E+02 |
| 4 | 0.7480786258634735E+02 | 0.6980899077051444E+02 | 0.7355562567868589E+02 | 0.8418344360135590E+02 |
| 5 | 0.9221266466053553E+02 | 0.1071786719672733E+03 | 0.1442114658669020E+03 | 0.2146134760517165E+03 |
| 6 | 0.9941197242066054E+02 | 0.1230668406481259E+03 | 0.1782491047509949E+03 | 0.2861457598164388E+03 |
| 7 | 0.1026103523087546E+03 | 0.1332258620524329E+03 | 0.1982329465874464E+03 | 0.3215568765099048E+03 |
| 8 | 0.1020281417107785E+03 | 0.1271624587073099E+03 | 0.1826056439986178E+03 | 0.2907755507659710E+03 |
| 9 | 0.9039603910129196E+02 | 0.1071887267216484E+03 | 0.1548665991025587E+03 | 0.2601480824700848E+03 |
| 10 | 0.8095873974206875E+02 | 0.9100740357495621E+02 | 0.1252754610703735E+03 | 0.1946469657224905E+03 |
| 11 | 0.6921123568145022E+02 | 0.6960486142219685E+02 | 0.8734552577042620E+02 | 0.1250776911481737E+03 |
| 12 | 0.6486261053527244E+02 | 0.6318561771868312E+02 | 0.7713336903367377E+02 | 0.1068738254836624E+03 |
| 13 | 0.8754097356965494E+02 | 0.9914268128277373E+02 | 0.1325186914190332E+03 | 0.1965513339618076E+03 |
| 14 | 0.1209721959332602E+03 | 0.1809394580047123E+03 | 0.3184685694287427E+03 | 0.6365253220760596E+03 |
| 15 | 0.1588109522148007E+03 | 0.3190952114017182E+03 | 0.7462518429749929E+03 | 0.1945497662969014E+04 |
| 16 | 0.1906974609084911E+03 | 0.4719326683354067E+03 | 0.1390009252743353E+04 | 0.4668645348355675E+04 |
| 17 | 0.2357465080233630E+03 | 0.7096677571992062E+03 | 0.2535041781642134E+04 | 0.1017905969259994E+05 |
| 18 | 0.2752879991704121E+03 | 0.9370821562558177E+03 | 0.3696764032756442E+04 | 0.1615808119903336E+05 |
| 19 | 0.3101276851917939E+03 | 0.1187907352689284E+04 | 0.5217290405067348E+04 | 0.2523229260765624E+05 |
| 20 | 0.3698634891445204E+03 | 0.1780382583172881E+04 | 0.1030534549859066E+05 | 0.6886292721027552E+05 |
| 21 | 0.4176967218993768E+03 | 0.2234484799909078E+04 | 0.1425255714990113E+05 | 0.1038587526780708E+06 |
| 22 | 0.1333645961824457E+04 | 0.1849978128888260E+05 | 0.2669081121982264E+06 | 0.4003689372775944E+07 |
| 23 | 0.6989573530171479E+03 | 0.5610648162623282E+04 | 0.5076928191332528E+05 | 0.5120438850649450E+06 |
| 24 | 0.4042855384246694E+03 | 0.1999407580366985E+04 | 0.1158400762057241E+05 | 0.7690222047253189E+05 |
| 25 | 0.3311545045919114E+03 | 0.1311440372949007E+04 | 0.5965381978401817E+04 | 0.3044522760629125E+05 |
| 26 | 0.3373086709604516E+03 | 0.1391019268494739E+04 | 0.6622236919095653E+04 | 0.3521635642419534E+05 |
| 27 | 0.2677625104741695E+03 | 0.8982313026748155E+03 | 0.3520912008497006E+04 | 0.1548092050558773E+05 |
| 28 | 0.2327844363369372E+03 | 0.7388495430447814E+03 | 0.2899636668889050E+04 | 0.1311269163592274E+05 |
| 29 | 0.1969748167184020E+03 | 0.5114305967197694E+03 | 0.1562306019027880E+04 | 0.5278632129567480E+04 |
| 30 | 0.1366760730506631E+03 | 0.2326270595902629E+03 | 0.4569588092572559E+03 | 0.1004100888186820E+04 |
| 31 | 0.1184996267907274E+03 | 0.1777867999314246E+03 | 0.3123635125209154E+03 | 0.6252632696693965E+03 |
| 32 | 0.8700976947091932E+02 | 0.1029713688274607E+03 | 0.1450598111534421E+03 | 0.2273903600923206E+03 |
| 33 | 0.6506228822272112E+02 | 0.6585478505946402E+02 | 0.8528614697432775E+02 | 0.1260397997430225E+03 |

```
MODULATION TECHNIQUE = MSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =             1
BIT RATE =  0.1200000000000000E+04
CARRIER FREQUENCY =       3000 HZ
MAXIMUM CARRIER AMPLITUDE =             1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.1111111111111111E-03 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES          SUM OF VARIANCES**2
 0.1525890150844788E+03   0.2084564025759793E+04
   SUM OF SKEWNESS            SUM OF SKEWNESS**2
-0.1665578258850635E+05   0.9704853819233804E+08
   SUM OF KURTOSIS            SUM OF KURTOSIS**2
 0.5423666237610907E+04   0.6118691548293858E+07
    MEAN VARIANCE          VARIANCE OF THE VARIANCES
 0.4623909548014508E+01   0.4309394443714190E+02
    MEAN SKEWNESS          VARIANCE OF THE SKEWNESS
-0.5047206845001924E+03   0.2770063131355647E+07
    MEAN KURTOSIS          VARIANCE OF THE KURTOSIS
 0.1643535223518457E+03   0.1633529030650538E+06
```



0.000E+00
1.778E-03
3.556E-03
5.333E-03
7.111E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50

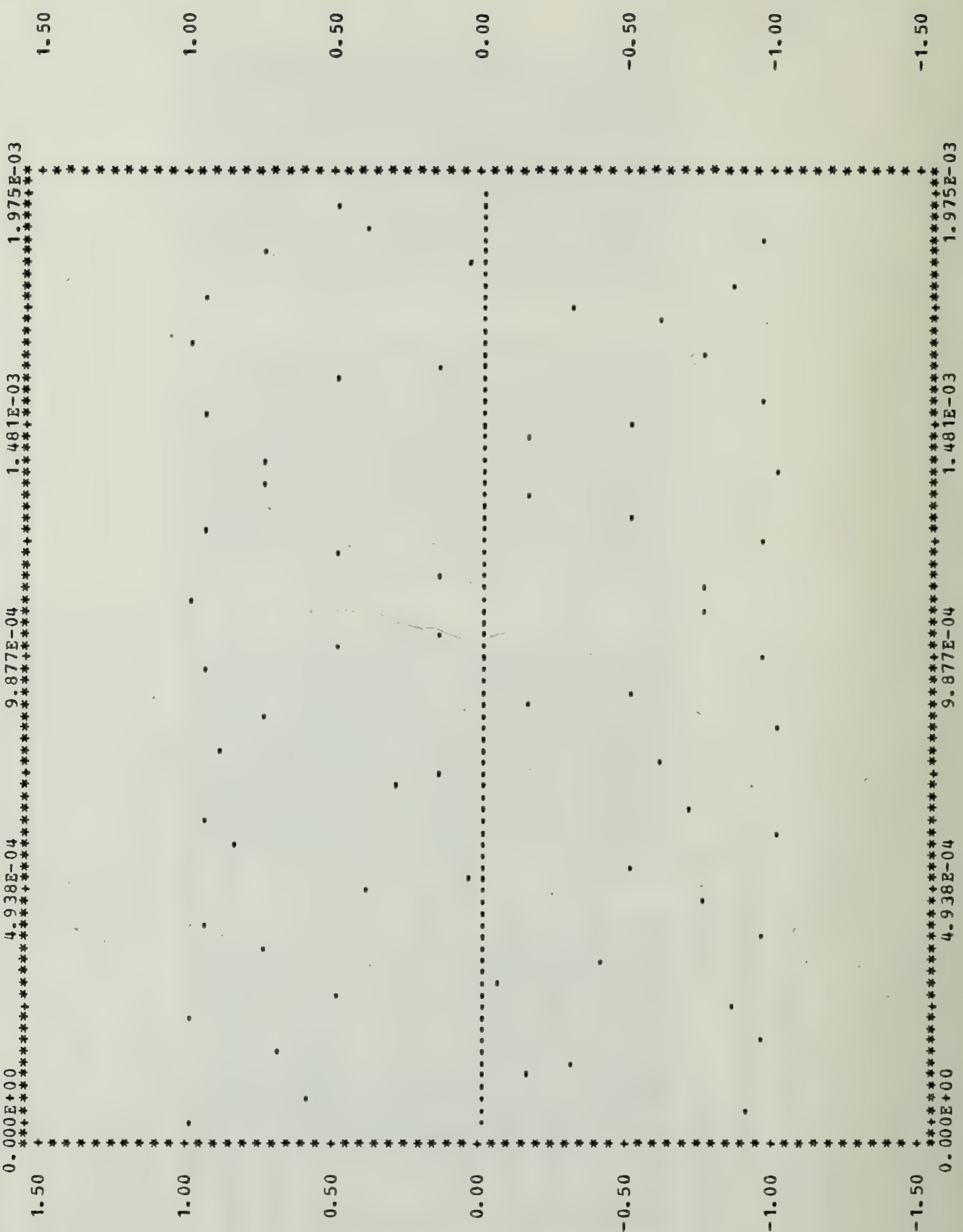


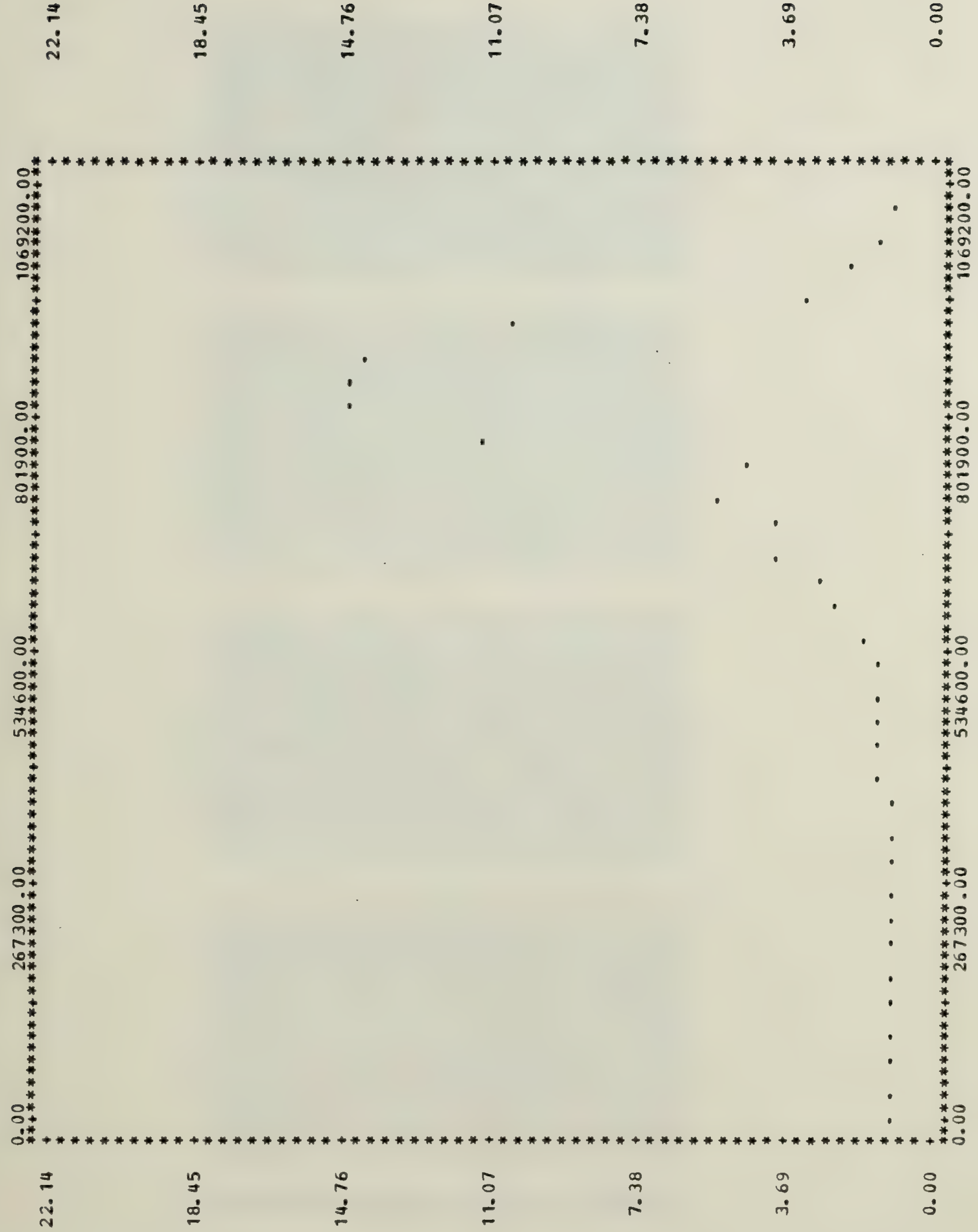
0.00
74250.00
148500.00
222750.00
297000.00
19.28
16.07
12.85
9.64
6.43
3.21
0.00

| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.1745381057261094E+01 | 0.1355578995944516E+01 | -0.3424362342353158E+01 | 0.5473118008703764E+01 |
| 2 | 0.1670893117553454E+01 | 0.5901911630020479E+00 | -0.3495315053453071E+01 | 0.1090755818972799E+01 |
| 3 | 0.1549301951756876E+01 | 0.6168607541593016E+00 | -0.2050204688587283E+01 | 0.2072568134087788E+01 |
| 4 | 0.1661199892949759E+01 | 0.6791172245377799E+00 | -0.3416316593482210E+01 | 0.1031766180903137E+01 |
| 5 | C.1789165204967162E+01 | 0.1187649235767541E+01 | -0.4253824550585361E+01 | 0.4425773685632588E+01 |
| 6 | 0.1793980989469851E+01 | 0.6705838354600710E+00 | -0.5135274827787229E+01 | 0.9492102864198628E+00 |
| 7 | 0.1744198082222983E+01 | 0.8410759608141099E+00 | -0.4231013557271023E+01 | 0.1621868551827440E+01 |
| 8 | 0.1974720124125793E+01 | 0.9695494611582597E+00 | -0.7310166411774525E+01 | 0.2277449619040244E+01 |
| 9 | 0.2053488485398217E+01 | 0.9611806657981106E+00 | -0.8487669792860372E+01 | 0.2604980667152322E+01 |
| 10 | 0.2088844746733196E+01 | 0.8767072712879614E+00 | -0.9051519777937444E+01 | 0.2339329653500119E+01 |
| 11 | 0.2078335081298121E+01 | 0.8586827233682362E+00 | -0.9240745527458049E+01 | 0.1528701296758548E+01 |
| 12 | 0.1846376751847116E+01 | 0.1102239221695370E+01 | -0.5290548653695978E+01 | 0.3019964469265048E+01 |
| 13 | 0.2245946499482377E+01 | 0.1194857291064416E+01 | -0.1223253783690999E+02 | 0.3015304601135153E+01 |
| 14 | 0.2963675540203763E+01 | 0.2445326920526502E+01 | -0.3299670208754627E+02 | 0.1525288363748558E+02 |
| 15 | 0.3173784304747806E+01 | 0.2467783740162910E+01 | -0.4212913869181091E+02 | 0.1518478689342305E+02 |
| 16 | 0.3465632704745810E+01 | 0.5120393955377951E+01 | -0.4209380992274822E+02 | 0.1417059143970117E+03 |
| 17 | 0.4614191970595573E+01 | 0.4884566208791112E+01 | -0.1448217452612039E+03 | 0.8434809081291280E+02 |
| 18 | 0.6056467485580745E+01 | 0.1025151080329688E+02 | -0.3488271204826219E+03 | 0.2873969971148171E+03 |
| 19 | 0.6406000917972383E+01 | 0.7829202751917509E+01 | -0.4368456489015806E+03 | 0.1481115108052472E+03 |
| 20 | 0.1698791308733019E+02 | 0.2854008167165727E+02 | -0.9245547474814263E+04 | 0.2008740923582969E+04 |
| 21 | 0.6683213299026370E+01 | 0.1119947335159933E+02 | -0.4907458525202289E+03 | 0.3031919538307138E+03 |
| 22 | 0.6486578159166406E+01 | 0.1017117379541286E+02 | -0.4465802292353152E+03 | 0.2680409342301182E+03 |
| 23 | 0.6465666718341273E+01 | 0.7377630480256458E+01 | -0.4512761578680025E+03 | 0.1522694237588385E+03 |
| 24 | 0.1116454812650914E+02 | 0.2162335800890371E+02 | -0.2501564247125116E+04 | 0.1165597750187054E+04 |
| 25 | 0.1044373719208894E+02 | 0.1462960849421151E+02 | -0.2068981176065553E+04 | 0.6248220438872377E+03 |
| 26 | 0.4981051119955246E+01 | 0.6452035401677616E+01 | -0.1905233821159606E+03 | 0.1076197803024863E+03 |
| 27 | 0.4123269596834735E+01 | 0.4439736504910233E+01 | -0.1001918569207450E+03 | 0.6134152643541944E+02 |
| 28 | C.2864935711027876E+01 | 0.1615487880606123E+01 | -0.3017501863379686E+02 | 0.5746676318150122E+01 |
| 29 | 0.1801795405750350E+01 | 0.9106026395844705E+00 | -0.4741745708792403E+01 | 0.2331085528965168E+01 |
| 30 | 0.1309968810623288E+01 | 0.3731036902683101E+00 | -0.9703918339010844E+00 | 0.3451406134067727E+00 |
| 31 | 0.8612683866837359E+00 | 0.2253789614502914E+00 | 0.2871099110118729E+00 | 0.1541083993062977E+00 |
| 32 | 0.6043391628396291E+00 | 0.7921584690864393E-01 | 0.2862143793099431E+00 | 0.1130792746883191E-01 |
| 33 | 0.5680265238925039E+00 | 0.4907017290140923E-01 | 0.2752850066625604E+00 | 0.2607974479555311E-02 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.1745381057261094E+03 | 0.4388378241030927E+03 | 0.1346110434953155E+04 | 0.4708163227179921E+04 |
| 2 | 0.1670893117553454E+03 | 0.3376173061659528E+03 | 0.7844590323160757E+03 | 0.2034919190393599E+04 |
| 3 | 0.1549301951756876E+03 | 0.3011028684335374E+03 | 0.7144100088287587E+03 | 0.2026599445555989E+04 |
| 4 | 0.1661199892949759E+03 | 0.3431911136628693E+03 | 0.8167784476187219E+03 | 0.2132708178344390E+04 |
| 5 | 0.1789165204967162E+03 | 0.4376884874075052E+03 | 0.1283686155661075E+04 | 0.4297079886762252E+04 |
| 6 | 0.1793980989469851E+03 | 0.3882245787684695E+03 | 0.9322015009721566E+03 | 0.2394998604752789E+04 |
| 7 | 0.1744198082222983E+03 | 0.3874892151236301E+03 | 0.9960270919690049E+03 | 0.2814823675876644E+04 |
| 8 | 0.1974720124125794E+03 | 0.4859373535174067E+03 | 0.1367850258262828E+04 | 0.4224573233856694E+04 |
| 9 | 0.2053488485398217E+03 | 0.5168383818803192E+03 | 0.1491835026790595E+04 | 0.4772349863793718E+04 |
| 10 | 0.2088844746733196E+03 | 0.5231212574529951E+03 | 0.1500350818620739E+04 | 0.4786253224795818E+04 |
| 11 | 0.2078335081298121E+03 | 0.5169572606289023E+03 | 0.1435261336113773E+04 | 0.4284140702659404E+04 |
| 12 | 0.1846376751847116E+03 | 0.4500323939239922E+03 | 0.1281911761836188E+04 | 0.4050916330073661E+04 |
| 13 | 0.2245946499482377E+03 | 0.6227184396690914E+03 | 0.1963667979815342E+04 | 0.6729137619289132E+04 |
| 14 | 0.2963675540203763E+03 | 0.1120424635892330E+04 | 0.4905380513832378E+04 | 0.2377476116996190E+05 |
| 15 | 0.3173784304747807E+03 | 0.1251601271582480E+04 | 0.5689442182858819E+04 | 0.2854218554507767E+05 |
| 16 | 0.3465632704745810E+03 | 0.1707980006002793E+04 | 0.1114619110245010E+05 | 0.8887814430949281E+05 |
| 17 | 0.4614191970595573E+03 | 0.2612648808821186E+04 | 0.1742546143252428E+05 | 0.1322897499994962E+06 |
| 18 | 0.6056467485580745E+03 | 0.4682979409916067E+04 | 0.4286806586636075E+05 | 0.4402476375446830E+06 |
| 19 | 0.6406000917972383E+03 | 0.4878775848546135E+04 | 0.4186839325073341E+05 | 0.3915963851799988E+06 |
| 20 | 0.1698791308733019E+04 | 0.3168438719176251E+05 | 0.6324822618229748E+06 | 0.1330157521040380E+08 |
| 21 | 0.6683213299026370E+03 | 0.5575281861836628E+04 | 0.5377474040246168E+05 | 0.5722380556849635E+06 |
| 22 | C.6486578159166406E+03 | 0.5214515827243337E+04 | 0.4839993126034607E+05 | 0.4972868019579191E+06 |
| 23 | 0.6465666718341274E+03 | 0.4910870028811991E+04 | 0.4176756170083457E+05 | 0.3879520293056626E+06 |
| 24 | 0.1116454812650914E+04 | 0.1460542592979535E+05 | 0.2141030904177548E+06 | 0.3415942298203231E+07 |
| 25 | 0.1044373719208894E+04 | 0.1235549589466912E+05 | 0.1584002090915345E+06 | 0.2162824361492207E+07 |
| 26 | 0.4981051119955246E+03 | 0.3119838530726825E+04 | 0.2260571335915158E+05 | 0.1814011642212024E+06 |
| 27 | 0.4123269596834735E+03 | 0.2139669130804281E+04 | 0.1304784189728246E+05 | 0.8978341400784285E+05 |
| 28 | 0.2864935711027876E+03 | 0.9807189630122864E+03 | 0.3770017149804132E+04 | 0.1569116396929776E+05 |
| 29 | 0.1801795405750350E+03 | 0.4147963297371694E+03 | 0.1118666456011763E+04 | 0.3377669865106972E+04 |
| 30 | 0.1309968810623288E+03 | 0.2085390938171418E+03 | 0.3792962857370318E+03 | 0.7582535161578269E+03 |
| 31 | 0.8612683866837359E+02 | 0.9649084057365937E+02 | 0.1296678760929258E+03 | 0.1977471929733572E+03 |
| 32 | 0.6043391628396291E+02 | 0.4436495121812613E+02 | 0.3601070561840355E+02 | 0.3097912706858261E+02 |
| 33 | 0.5680265238925039E+02 | 0.3712336030177964E+02 | 0.2625883421946238E+02 | 0.1928732784350291E+02 |

```
MODULATION TECHNIQUE = MFSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =            3
BIT RATE =  0.360000000000000E+04
CARRIER FREQUENCY =       10800 HZ
MAXIMUM CARRIER AMPLITUDE =            1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.308641975308641 9E-04 SEC
NUMBER OF SAMPLES GENERATED =           64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.4806181141925261E+03   0.1352273679574422E+05
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.1654247621695240E+04   0.5890267815807785E+06
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.5341287673934422E+05   0.2790612658473240E+09
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.1456418527856140E+02   0.2038414228878589E+03
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.5012871580894667E+02   0.1581567102109974E+05
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.1618572022404370E+04   0.6019021185027759E+07
```

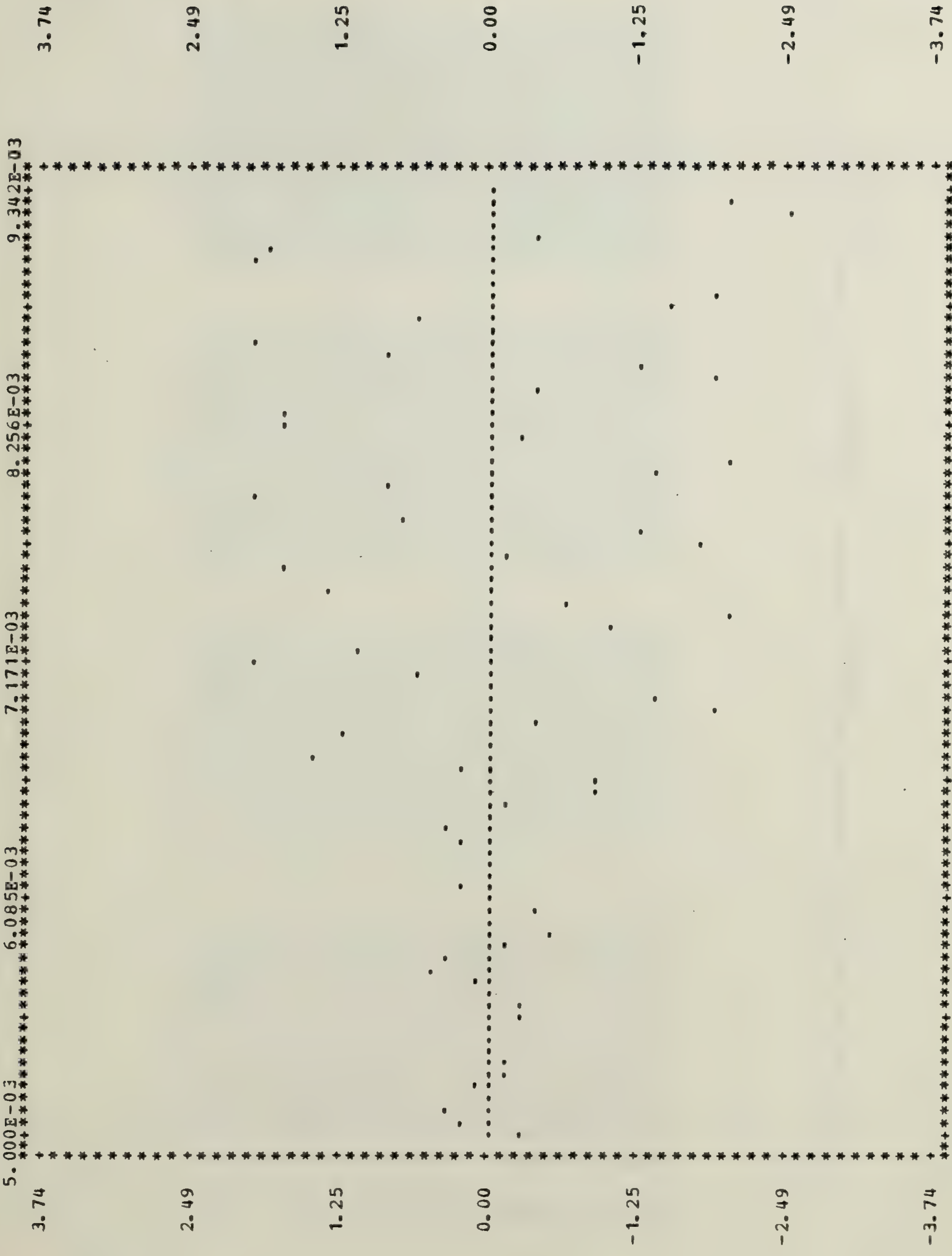

0.000E+00
4.938E-04
9.877E-04
1.481E-03
1.975E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50

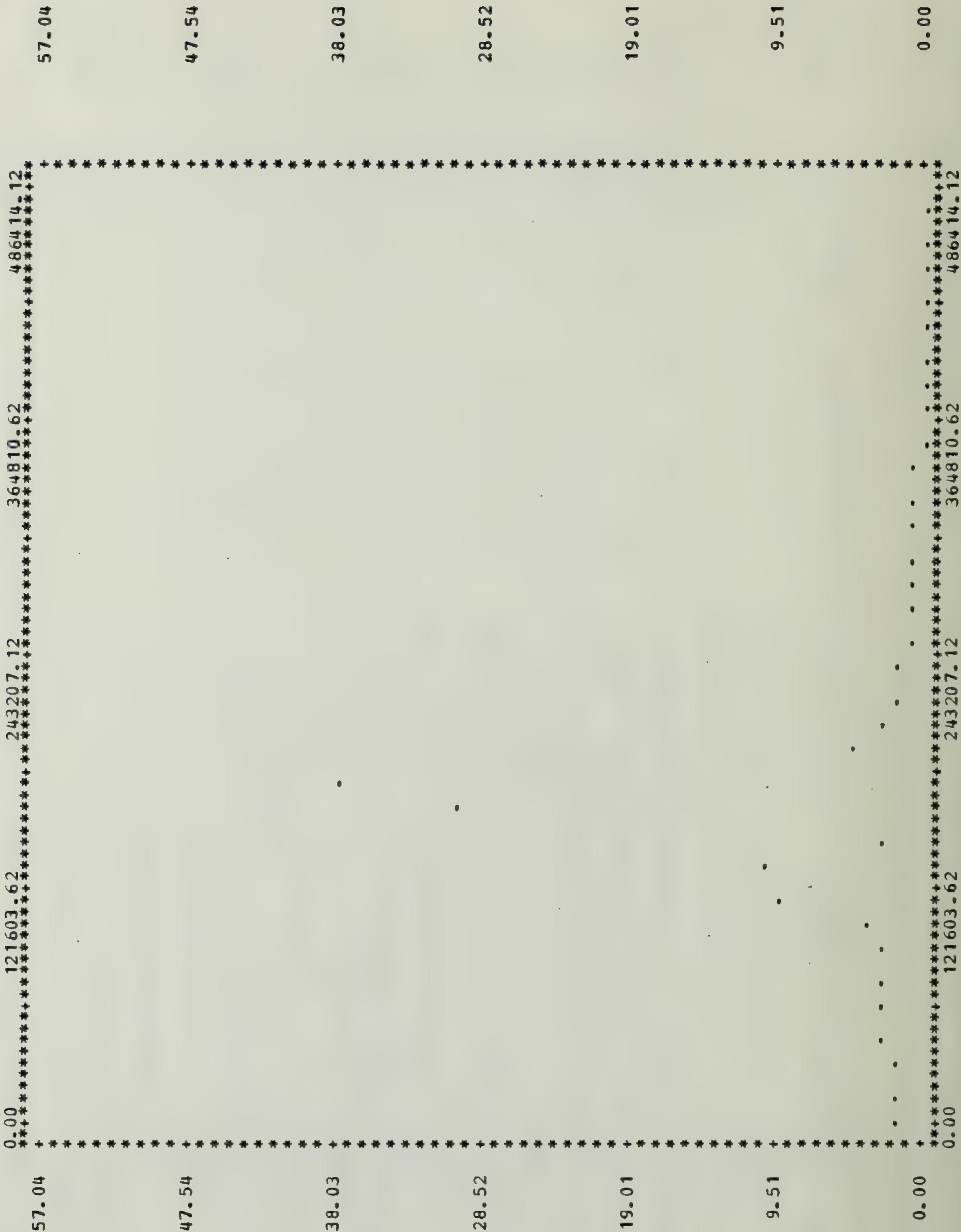

0.00
267300.00
534600.00
801900.00
1069200.00
22.14
18.45
14.76
11.07
7.38
3.69
0.00

| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.5772242127037718E+00 | 0.1701870281944493E+00 | 0.2881221581889301E+00 | 0.5202216579936257E-01 |
| 2 | 0.5802219943279051E+00 | C.1692179389704175E+00 | 0.2881870091118721E+00 | 0.4962377142977341E-01 |
| 3 | 0.6028101560497944E+00 | 0.1630844574986009E+00 | 0.2975209809667937E+00 | 0.4759923908949447E-01 |
| 4 | 0.6236957775C89775E+00 | 0.1517433865402484E+00 | 0.3011297117696356E+00 | 0.4060584454262756E-01 |
| 5 | 0.6603884574738126E+00 | 0.1541310275696224E+00 | 0.3222793291193875E+00 | 0.5372574080695565E-01 |
| 6 | 0.6914911473735869E+00 | 0.1625277259401340E+00 | 0.3303730158038245E+00 | 0.6338732183899803E-01 |
| 7 | 0.6712915135223571E+00 | 0.1989459204647430E+00 | 0.3457384652309939E+00 | 0.9079440007191118E-01 |
| 8 | 0.8082796112917950E+00 | 0.2158539044684256E+00 | 0.2513665218179688E+00 | 0.8711759471530689E-01 |
| 9 | 0.8347057408743954E+00 | 0.2063633947272559E+00 | 0.2416258755581653E+00 | 0.7897411259783627E-01 |
| 10 | 0.1155203641180799E+01 | 0.3253381023466752E+00 | -0.3616711703060194E+00 | 0.3009444434532169E+00 |
| 11 | 0.1405578654040239E+01 | 0.5901481998178570E+00 | -0.1545347282296657E+01 | 0.6485308087822852E+00 |
| 12 | 0.1653469595833664E+01 | 0.1006408909069761E+01 | -0.3264525934540579E+01 | 0.1839454693919151E+01 |
| 13 | 0.3282094270411689E+01 | 0.8604653034726285E+01 | -0.2794322254332445E+02 | 0.1742698305978736E+03 |
| 14 | 0.5652678805043730E+01 | 0.4459844156194543E+02 | 0.1192554149182590E+03 | 0.8446317926807378E+04 |
| 15 | 0.5302792986411789E+01 | 0.1999384407499615E+02 | -0.1915066362357811E+03 | 0.8114244563340110E+03 |
| 16 | 0.4839236365113163E+01 | 0.3116228371110405E+02 | 0.1153475974583148E+03 | 0.5030156057294299E+04 |
| 17 | 0.4799926056893745E+01 | 0.1627827150809915E+02 | -0.9300212955009531E+02 | 0.1027015195900844E+04 |
| 18 | 0.4973004323275781E+01 | 0.1660039482645575E+02 | -0.1210599846314087E+03 | 0.9267637549193778E+03 |
| 19 | 0.4949076398807320E+01 | 0.3195721379055708E+02 | 0.1586666023965808E+03 | 0.6953906279136898E+04 |
| 20 | 0.5240481051222312E+01 | 0.1710861900951835E+02 | -0.1294472248356874E+03 | 0.1518760678679011E+04 |
| 21 | 0.5676550996298115E+01 | 0.4183417025969290E+02 | 0.1323276886342935E+03 | 0.8939299980725979E+04 |
| 22 | 0.6101993574152731E+01 | 0.1635330658894298E+02 | -0.3215862356984157E+03 | 0.7932740986113413E+03 |
| 23 | 0.6259114577537281E+01 | 0.2770207472842645E+02 | -0.2779436555304097E+03 | 0.1949698968797135E+04 |
| 24 | 0.6284446589130466E+01 | 0.2662812884562886E+02 | -0.2772962242103021E+03 | 0.2129199108385613E+04 |
| 25 | 0.5742277487325188E+01 | 0.1282122464430269E+02 | -0.2785797147037739E+03 | 0.4954058112039018E+03 |
| 26 | 0.5170868203869401E+01 | 0.2976675343581255E+02 | -0.3721000245806551E+02 | 0.2486653253210702E+04 |
| 27 | 0.5917383281247872E+01 | 0.2149629222581870E+02 | -0.2452121102148178E+03 | 0.1728196939758512E+04 |
| 28 | 0.5541030766931423E+01 | 0.3540159389584892E+02 | -0.2604124431555488E+02 | 0.4059766140333669E+04 |
| 29 | 0.5288745936690563E+01 | 0.1787438125039946E+02 | -0.2028946595998615E+03 | 0.6603169983834629E+03 |
| 30 | 0.5074754282284268E+01 | 0.3131561713839394E+02 | 0.1173219615183484E+01 | 0.2905905006347744E+04 |
| 31 | 0.4253003494334969E+01 | 0.2577288744924743E+02 | 0.5235457234353824E+02 | 0.2345083299105233E+04 |
| 32 | 0.1937545429384470E+01 | 0.2626549296496931E+01 | -0.3005281031163669E+01 | 0.2094124943834442E+02 |
| 33 | 0.1137815044472511E+01 | 0.1207462920504103E+01 | 0.1860809816827437E+01 | 0.7168925235852774E+01 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.577224212703771 8E+02 | 0.5016729496439940E+02 | 0.4904798949050517E+02 | 0.5146241506306405E+02 |
| 2 | 0.5802219943279051E+02 | 0.5041833222825649E+02 | 0.4924866419931451E+02 | 0.5142222707838669E+02 |
| 3 | 0.6028101560497944E+02 | 0.5248336971603924E+02 | 0.5198860611495787E+02 | 0.5530202080413826E+02 |
| 4 | 0.6236957775089775E+02 | 0.5392223755573740E+02 | 0.5320690223250621E+02 | 0.5634238544784942E+02 |
| 5 | 0.6603884574738126E+02 | 0.5887026320585678E+02 | 0.6163707688778974E+02 | 0.7120408044653105E+02 |
| 6 | 0.6914911473735869E+02 | 0.6390624555767724E+02 | 0.6997710939622378E+02 | 0.8513957160811946E+02 |
| 7 | 0.6712915135223571E+02 | 0.6475887573872325E+02 | 0.7486363842363333E+02 | 0.9592693526928423E+02 |
| 8 | 0.8082796112917950E+02 | 0.8670112954537567E+02 | 0.1047097320382813E+03 | 0.1354378623407661E+03 |
| 9 | 0.8347057408743954E+02 | 0.9010334346286568E+02 | 0.1104451869673761E+03 | 0.1456174802950561E+03 |
| 10 | 0.1155203641180799E+03 | 0.1656580173920585E+03 | 0.2710400270962173E+03 | 0.4903643576994201E+03 |
| 11 | 0.1405578654040239E+03 | 0.2559898070513248E+03 | 0.5297764265612851E+03 | 0.1179902655990948E+04 |
| 12 | 0.1653469595833664E+03 | 0.3730306524325403E+03 | 0.9771395920102751E+03 | 0.2769882764951451E+04 |
| 13 | 0.3282094270411689E+03 | 0.1929074930424826E+04 | 0.1404546651472015E+05 | 0.1119512581698045E+06 |
| 14 | 0.5652678805043730E+03 | 0.7610523481931659E+04 | 0.1345945203020359E+06 | 0.2735139972972573E+07 |
| 15 | 0.5302792986411789E+03 | 0.4791351909098425E+04 | 0.5144805558206787E+05 | 0.6012440154264541E+06 |
| 16 | 0.4839236365113163E+03 | 0.5426886947142667E+04 | 0.8563708401825700E+05 | 0.1562683895259645E+07 |
| 17 | 0.4799926056693745E+03 | 0.3915477894466569E+04 | 0.4247394212718559E+05 | 0.5361723252608621E+06 |
| 18 | 0.4973004323275782E+03 | 0.4116516287751081E+04 | 0.4436220702445909E+05 | 0.5477857354242532E+06 |
| 19 | 0.4949076398807320E+03 | 0.5613099885388315E+04 | 0.9430696930018190E+05 | 0.1917395613042084E+07 |
| 20 | 0.5240481051222312E+03 | 0.4440017446764327E+04 | 0.5136623107738203E+05 | 0.7232624679783723E+06 |
| 21 | 0.5676550996298115E+03 | 0.7363905977066910E+04 | 0.1321928860530092E+06 | 0.2783296553532304E+07 |
| 22 | 0.6101993574152731E+03 | 0.5342409910205477E+04 | 0.5819256414204924E+05 | 0.7220823402526333E+06 |
| 23 | 0.6259114577537281E+03 | 0.6660156927588188E+04 | 0.8943038733047049E+05 | 0.1328899427697067E+07 |
| 24 | 0.6284446589130466E+03 | 0.6585611648880613E+04 | 0.8853229778507997E+05 | 0.1345801576503997E+07 |
| 25 | 0.5742277487325188E+03 | 0.4566676313930133E+04 | 0.4421664614948079E+05 | 0.4878553805504906E+06 |
| 26 | 0.5170868203869401E+03 | 0.5620696388324199E+04 | 0.7812306487180346E+05 | 0.1177282884327380E+07 |
| 27 | 0.5917383281247872E+03 | 0.5629675420075235E+04 | 0.6841454562789571E+05 | 0.9772311809625675E+06 |
| 28 | 0.5541030766931423E+03 | 0.6575059991697107E+04 | 0.1005531003016690E+06 | 0.1706205303321739E+07 |
| 29 | 0.5288745936690563E+03 | 0.4566647102075641E+04 | 0.4657187623964849E+05 | 0.5195716873974923E+06 |
| 30 | 0.5074754282284268E+03 | 0.5675559199257252E+04 | 0.8137290080877166E+05 | 0.1264367017074687E+07 |
| 31 | 0.4253003494334969E+03 | 0.4360319729758042E+04 | 0.5725121463102471E+05 | 0.8334021441901511E+06 |
| 32 | 0.1937545429384470E+03 | 0.6354366094460613E+03 | 0.2642217333585245E+04 | 0.1248684163529250E+05 |
| 33 | 0.1137815044472511E+03 | 0.2490011366727044E+03 | 0.7771080847880242E+03 | 0.2822352571960914E+04 |

```
MODULATION TECHNIQUE = QPRS
QPRS CLASS 1 FILTER
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =          1200 HZ
BITS PER BINARY CODE WORD =              2
BIT RATE =  0.2400000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =              1 VOLT(S)
INITIAL PHASE ANGLE =             0 DEGREES
TIME BETWEEN SAMPLES =  0.6784342273548912E-04 SEC
NUMBER OF SAMPLES GENERATED =           64
SEED FOR RANDOM NUMBER GENERATOR =            1
NUMBER OF TIMES SIMULATION REPEATS =          100
   SUM OF VARIANCES           SUM OF VARIANCES**2
 0.7925891201989905E+03    0.1512135267846902E+06
   SUM OF SKEWNESS             SUM OF SKEWNESS**2
-0.1732359062524069E+06    0.9850778236273860E+10
   SUM OF KURTOSIS             SUM OF KURTOSIS**2
 0.3533192079165041E+06    0.4536060891356758E+11
    MEAN VARIANCE           VARIANCE OF THE VARIANCES
 0.2401785212724214E+02    0.4130538703065308E+04
    MEAN SKEWNESS           VARIANCE OF THE SKEWNESS
-0.5249572916739602E+04    0.2794176160813870E+09
    MEAN KURTOSIS           VARIANCE OF THE KURTOSIS
 0.1070664266413649E+05    0.1299304575250932E+10
```

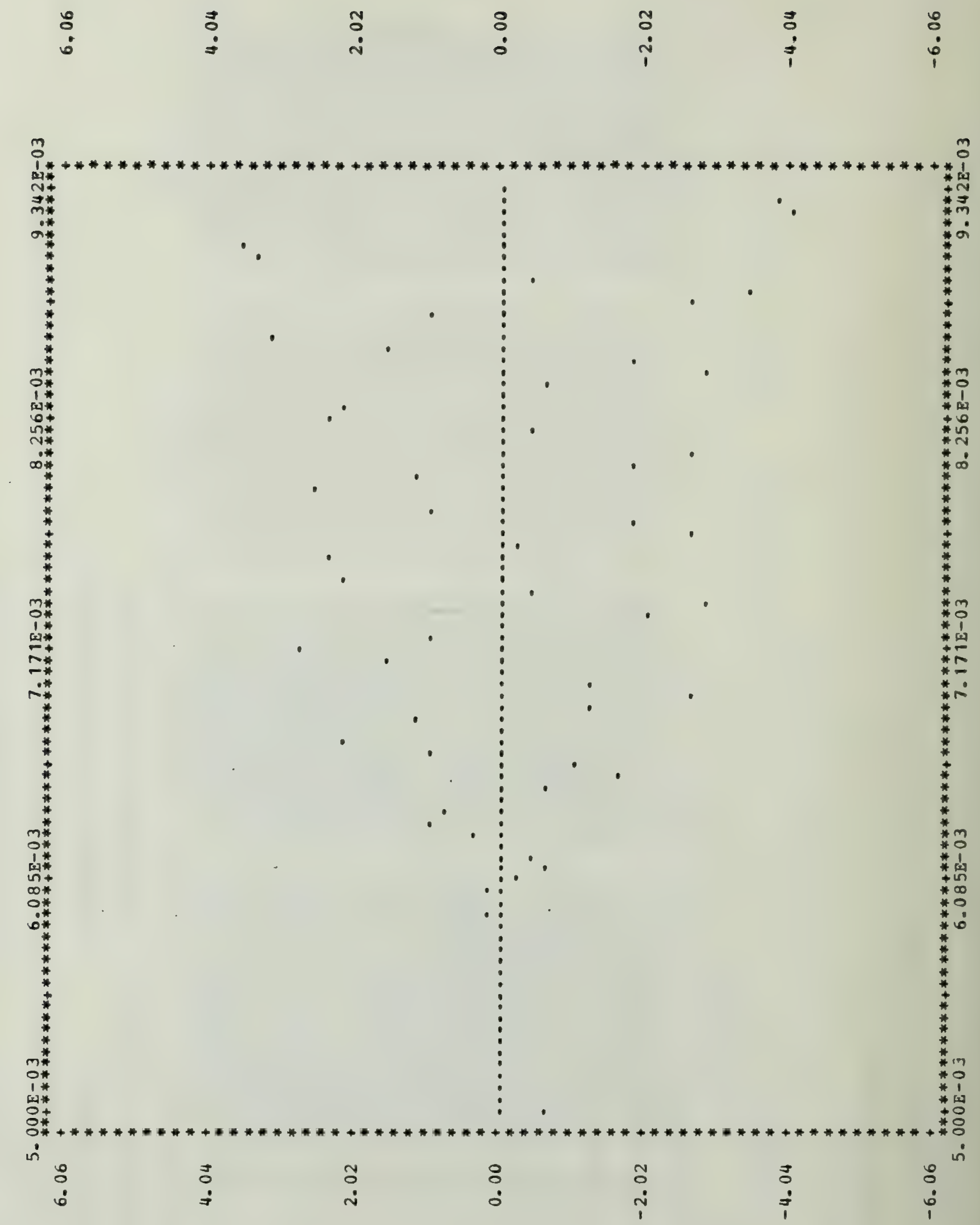

3.74
2.49
1.25
0.00
-1.25
-2.49
-3.74
5.000E-03
6.085E-03
7.171E-03
8.256E-03
9.342E-03


0.00
121603.62
243207.12
364810.62
486414.12
57.04
47.54
38.03
28.52
19.01
9.51
0.00

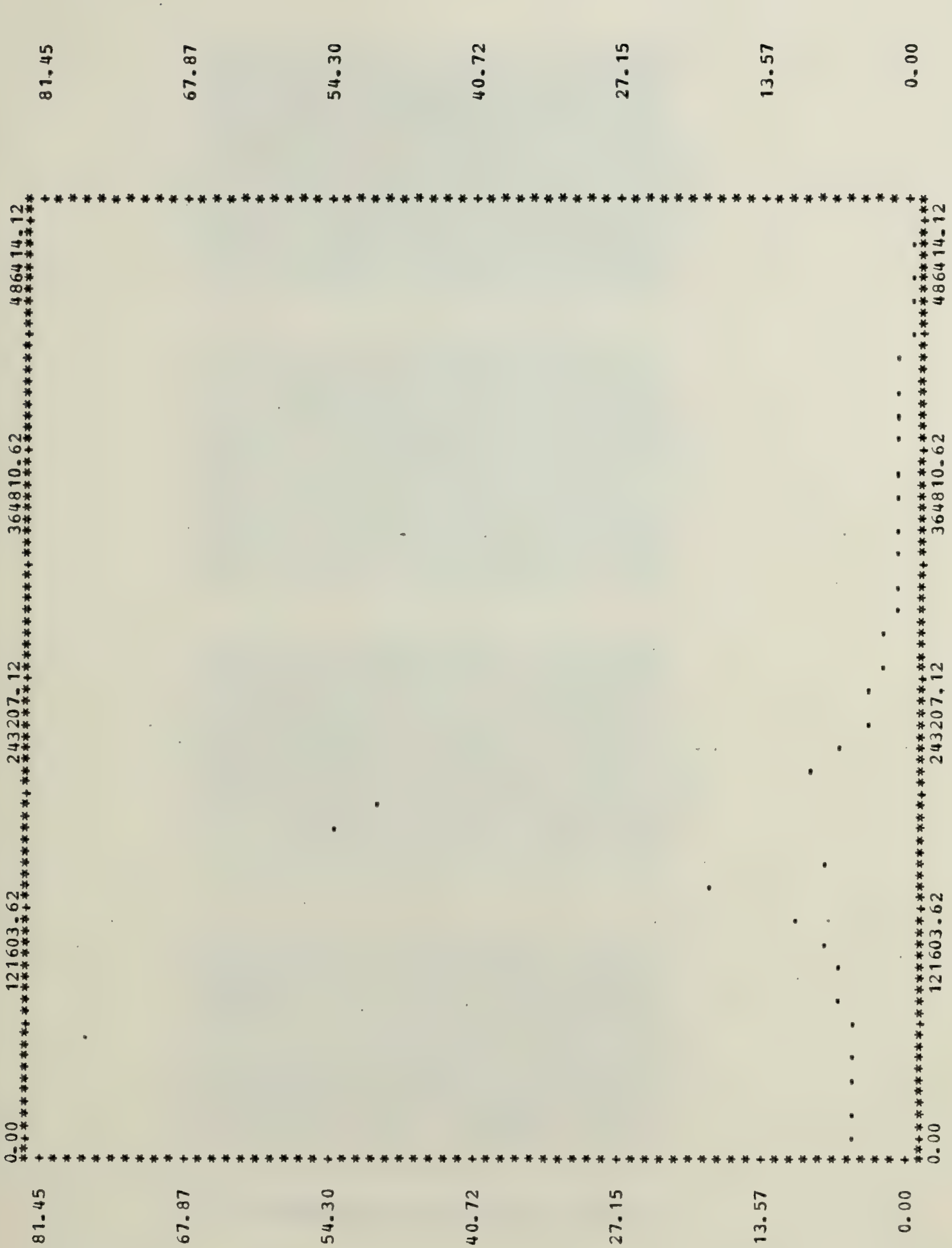
| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.1746811380402326E+01 | 0.1388348922506595E+01 | -0.3602282705486656E+01 | 0.5140926308007252E+01 |
| 2 | C.1783934079105657E+01 | 0.1373900592274223E+01 | -0.4000383957156492E+01 | 0.5076096043644750E+01 |
| 3 | 0.1881639570810958E+01 | 0.1391153714557888E+01 | -0.5219656911680200E+01 | 0.5267763272060726E+01 |
| 4 | 0.2045686771930169E+01 | 0.1468678508208007E+01 | -0.7699351745328097E+01 | 0.5884490676168223E+01 |
| 5 | 0.2297388544402002E+01 | 0.1644047467555237E+01 | -0.1261209753570515E+02 | 0.7239435166189224E+01 |
| 6 | 0.2683505678955374E+01 | 0.1999006273090073E+01 | -0.2311823186803285E+02 | 0.1024292414275401E+02 |
| 7 | 0.3312924180891911E+01 | 0.2756479367453149E+01 | -0.4950242176250788E+02 | 0.1834851180285642E+02 |
| 8 | 0.4514755520309362E+01 | 0.4811195957225893E+01 | -0.1408993176802396E+03 | 0.5414450274312225E+02 |
| 9 | 0.9536555141797188E+01 | 0.2327656842676565E+02 | -0.1505196860526031E+04 | 0.1409137635928702E+04 |
| 10 | 0.2195585557239151E+02 | 0.1129481209797817E+03 | -0.1990155013949577E+05 | 0.2735320445369462E+05 |
| 11 | 0.3232856430932140E+02 | 0.2402926487517075E+03 | -0.6527459761041918E+05 | 0.1283904731804624E+06 |
| 12 | 0.3329064024908647E+02 | 0.2623808601021426E+03 | -0.7047843429734775E+05 | 0.1648935745345857E+06 |
| 13 | 0.2012368604390696E+02 | 0.1051549869380090E+03 | -0.1496394469756025E+05 | 0.3059724135122297E+05 |
| 14 | 0.7502438388978744E+01 | 0.1396155254214945E+02 | -0.7105602094481494E+03 | 0.4730046115964915E+03 |
| 15 | 0.4016348508681620E+01 | 0.4194625326863587E+01 | -0.9514782451504106E+02 | 0.4603296277462294E+02 |
| 16 | 0.2970080794310893E+01 | 0.2376936951023241E+01 | -0.3349528176027253E+02 | 0.1564937982990288E+02 |
| 17 | 0.2390874625765719E+01 | 0.1616678403564954E+01 | -0.1494345472058580E+02 | 0.7763585773794798E+01 |
| 18 | 0.2016812253485666E+01 | 0.1211509773096788E+01 | -0.7479041281848517E+01 | 0.4629945683853585E+01 |
| 19 | 0.1753574112083879E+01 | 0.9704493989317589E+00 | -0.3959409356771873E+01 | 0.3114448386156138E+01 |
| 20 | 0.1558493513598559E+01 | 0.8148158249946477E+00 | -0.2125930435216646E+01 | 0.2271002271430383E+01 |
| 21 | 0.1408735774632585E+01 | 0.7084187564785975E+00 | -0.1102582833912087E+01 | 0.1754746256637649E+01 |
| 22 | 0.1290826693315262E+01 | 0.6327165569438174E+00 | -0.5020678660303746E+00 | 0.1417160155580148E+01 |
| 23 | 0.1196276693709804E+01 | 0.5773557484689719E+00 | -0.1363393565939444E+00 | 0.1185743805976271E+01 |
| 24 | 0.1119472616149126E+01 | 0.5361642063822957E+00 | 0.92643778905969C9E-01 | 0.1021682879331106E+01 |
| 25 | 0.1056567176528774E+01 | 0.5052688758882325E+00 | 0.2389420417312620E+00 | 0.9026889684670851E+00 |
| 26 | 0.1004854132567686E+01 | 0.4821360846597600E+00 | 0.3337377691836187E+00 | 0.8152494762041542E+00 |
| 27 | 0.9623989485715112E+00 | 0.4650482001544234E+00 | 0.3956861449378448E+00 | 0.7508171701876838E+00 |
| 28 | C.9278127756762294E+00 | 0.4528007126313904E+00 | 0.4362793202635347E+00 | 0.7038029740335077E+00 |
| 29 | 0.9001115711570695E+00 | 0.4445161706018519E+00 | 0.4627541658592416E+00 | 0.6704545675391549E+00 |
| 30 | 0.8786288878768683E+00 | 0.4395214050389312E+00 | 0.4797259791699269E+00 | 0.6481934210539040E+00 |
| 31 | 0.8629671538189785E+00 | 0.4372540540716741E+00 | 0.4901293234968129E+00 | 0.6351869105735284E+00 |
| 32 | 0.8529998029052858E+00 | 0.4371419567298135E+00 | 0.4957619571123395E+00 | 0.6299589957549181E+00 |
| 33 | 0.8490527574548856E+00 | 0.4382132490394413E+00 | 0.4975782018316388E+00 | 0.6304885575454902E+00 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.1746811380402326E+03 | 0.4425815431984608E+03 | 0.1348821158956003E+04 | 0.4629031129955732E+04 |
| 2 | 0.1783934079105657E+03 | 0.4542582384946031E+03 | 0.1394577701660524E+04 | 0.4823443801265942E+04 |
| 3 | 0.1881639570810958E+03 | 0.4917809651853957E+03 | 0.1545984386736991E+04 | 0.5476301191979648E+04 |
| 4 | 0.2045686771930169E+03 | 0.5638826091976001E+03 | 0.1853679535368173E+04 | 0.6851959557828961E+04 |
| 5 | 0.2297388544402002E+03 | 0.6905601116829235E+03 | 0.2442646091043520E+04 | 0.9659305428541077E+04 |
| 6 | 0.2683505678955374E+03 | 0.9180218939344913E+03 | 0.3638487164735292E+04 | 0.1597192190773616E+05 |
| 7 | 0.3312924180891911E+03 | 0.1370438120211696E+04 | 0.6475137258677694E+04 | 0.3353247956116903E+05 |
| 8 | 0.4514755520309362E+03 | 0.2514610140581750E+04 | 0.1589201479119533E+05 | 0.1095168622809752E+06 |
| 9 | 0.9536555141797188E+03 | 0.1139896867150364E+05 | 0.1574118170391285E+06 | 0.2406790106482331E+07 |
| 10 | 0.2195585557239151E+04 | 0.5938782336856992E+05 | 0.1825164485184334E+07 | 0.6097112730194072E+08 |
| 11 | 0.3232856430932141E+04 | 0.1283025792566120E+06 | 0.5707027578545261E+07 | 0.2739699803410351E+09 |
| 12 | 0.3329064024908647E+04 | 0.1368023779695217E+06 | 0.6393219475235099E+07 | 0.3266222349853693E+09 |
| 13 | 0.2012368604390696E+04 | 0.5090661770623646E+05 | 0.1495899359077720E+07 | 0.4897850625075236E+08 |
| 14 | 0.7502438388978744E+03 | 0.7010851879714993E+04 | 0.7548211554455331E+05 | 0.8952526070200140E+06 |
| 15 | 0.4016348508681620E+03 | 0.2028373441678403E+04 | 0.1169897042246145E+05 | 0.7429632439420937E+05 |
| 16 | 0.2970080794310893E+03 | 0.1117454750624743E+04 | 0.4842988519051916E+04 | 0.2330127244219820E+05 |
| 17 | 0.2390874625765719E+03 | 0.7316793095659672E+03 | 0.2610458719032500E+04 | 0.1044932950102081E+05 |
| 18 | 0.2016812253485666E+03 | 0.5266926341175751E+03 | 0.1625310013580066E+04 | 0.5684185523967754E+04 |
| 19 | 0.1753574112083879E+03 | 0.4035767071513206E+03 | 0.1112159628710498E+04 | 0.3503146647320578E+04 |
| 20 | 0.1558493513598559E+03 | 0.3235569698673484E+03 | 0.8144108665461795E+03 | 0.2358656873687845E+04 |
| 21 | 0.1408735774632585E+03 | 0.2685871051643480E+03 | 0.6279392110129216E+03 | 0.1697266410506734E+04 |
| 22 | 0.1290826693315262E+03 | 0.2292622943549593E+03 | 0.5043601709341667E+03 | 0.1286755414806408E+04 |
| 23 | 0.1196276693709804E+03 | 0.2002660118897543E+03 | 0.4188711664557241E+03 | 0.1017735231758019E+04 |
| 24 | 0.1119472616149126E+03 | 0.1784021502626242E+03 | 0.3577695558724314E+03 | 0.8339267628334791E+03 |
| 25 | 0.1056567176528774E+03 | 0.1616550385647335E+03 | 0.3130255874735007E+03 | 0.7042934667581610E+03 |
| 26 | 0.1004854132567686E+03 | 0.1487046551551519E+03 | 0.2797068731648117E+03 | 0.6107399469756730E+03 |
| 27 | 0.9623989485715112E+02 | 0.1386609454364429E+03 | 0.2546677115393676E+03 | 0.5422335522854365E+03 |
| 28 | 0.9278127756762294E+02 | 0.1309109252213106E+03 | 0.2358431093725300E+03 | 0.4918077124087745E+03 |
| 29 | 0.9001115711570695E+02 | 0.1250271849426682E+03 | 0.2218504961080261E+03 | 0.4549510898295330E+03 |
| 30 | 0.8786288878768683E+02 | 0.1207114913600284E+03 | 0.2117566636175042E+03 | 0.4287039457588381E+03 |
| 31 | 0.8629671538189785E+02 | 0.1177593822101386E+03 | 0.2049379073396893E+03 | 0.4111351799552002E+03 |
| 32 | 0.8529998029052858E+02 | 0.1160379200918972E+03 | 0.2009954318637255E+03 | 0.4010352890071484E+03 |
| 33 | 0.8490527574548856E+02 | 0.1154721701490792E+03 | 0.1997055966179411E+03 | 0.3977393746428140E+03 |

```
MODULATION TECHNIQUE = QPRS
QPRS CLASS 2 FILTER
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           2
BIT RATE =  0.24000000000000000E+04
CARRIER FREQUENCY =       2400 HZ
MAXIMUM CARRIER AMPLITUDE =            1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.678434227354891 2E-04 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.2406301297164022E+04   0.1712404031156036E+07
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.1014071770004887E+07   0.4097280851254902E+12
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.4062278349001294E+07   0.7316406994491162E+13
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.7291822112618248E+02   0.4802940065853782E+05
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.3072944757590566E+05   0.1183019436971622E+11
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.1230993439091301E+06   0.2130107248422762E+12
```



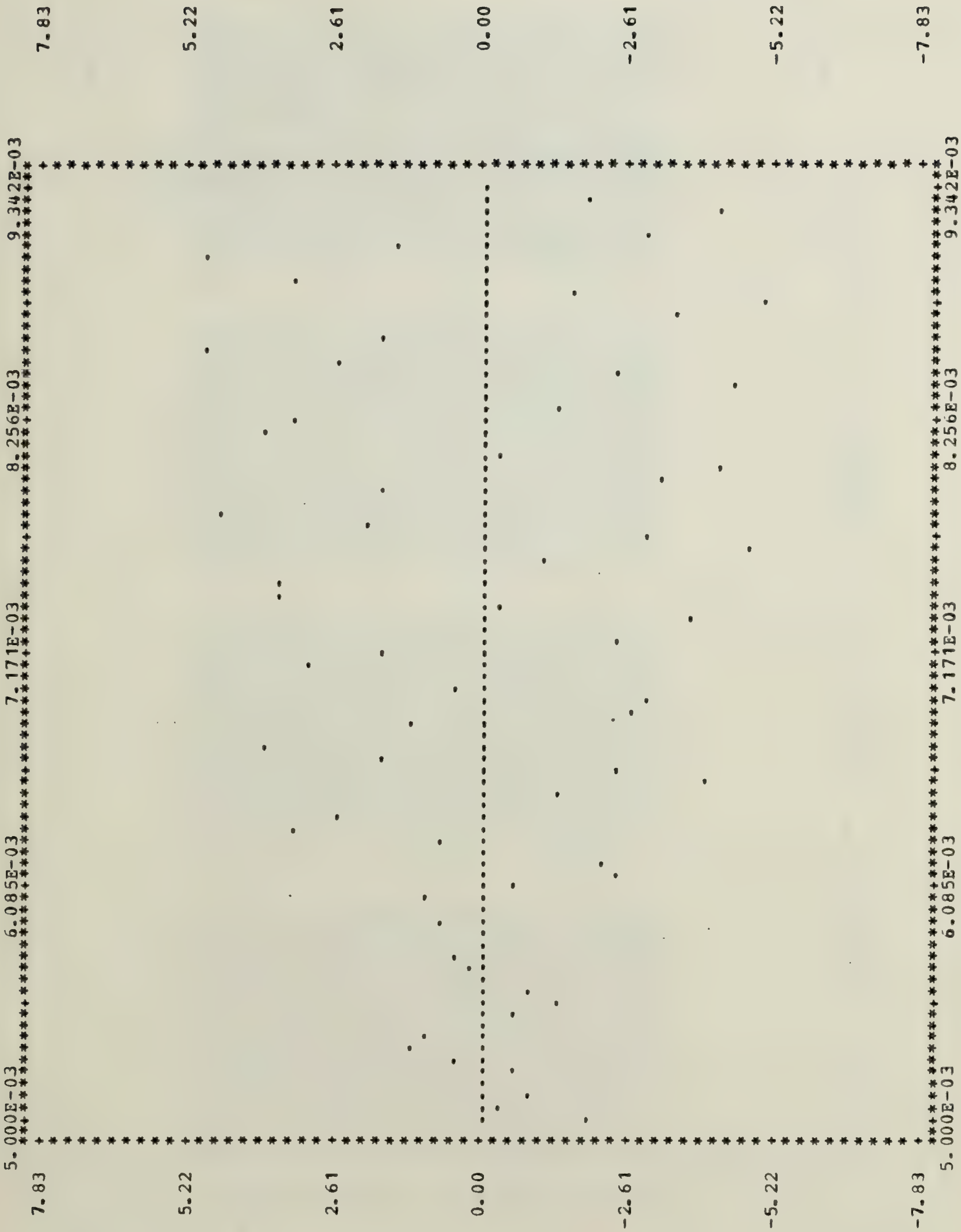
6.06
4.04
2.02
0.00
-2.02
-4.04
-6.06
5.000E-03
6.085E-03
7.171E-03
8.256E-03
9.342E-03

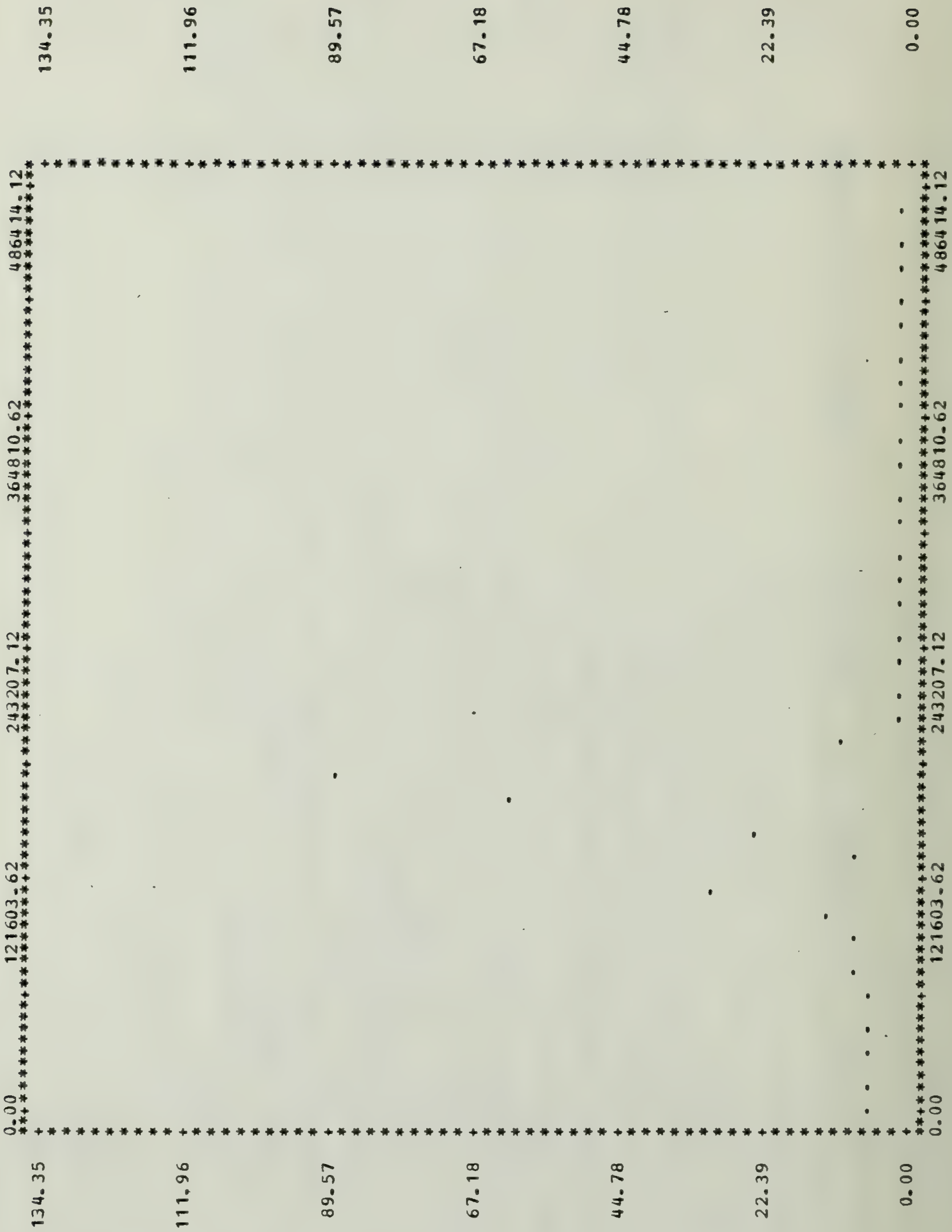

0.00
121603.62
243207.12
364810.62
486414.12
81.45
67.87
54.30
40.72
27.15
13.57
0.00

| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.2883744276236732E+01 | 0.3992201491926921E+01 | -0.2532518841826497E+02 | 0.4693293025936265E+02 |
| 2 | 0.2945575697221625E+01 | 0.3963052008893856E+01 | -0.2762424375864902E+02 | 0.4671417524690146E+02 |
| 3 | 0.3112787561988349E+01 | 0.4013664962406092E+01 | -0.3461233461524047E+02 | 0.4850914397069039E+02 |
| 4 | 0.3392105668005165E+01 | 0.4231239933942908E+01 | -0.4843388090938981E+02 | 0.5378883312445481E+02 |
| 5 | 0.3814330994522609E+01 | 0.4728891936605361E+01 | -0.7479410396850161E+02 | 0.6524120630247199E+02 |
| 6 | 0.4448304769119560E+01 | 0.5729652172852198E+01 | -0.1284525151380006E+03 | 0.8996407668008533E+02 |
| 7 | 0.5446496163994199E+01 | 0.7783957150351074E+01 | -0.2538211393046406E+03 | 0.1516517560198072E+03 |
| 8 | 0.7211506188528557E+01 | 0.1268735131795587E+02 | -0.6307639983631189E+03 | 0.3676854627938201E+03 |
| 9 | 0.1182998791891019E+02 | 0.3364874592461898E+02 | -0.2975891422630522E+04 | 0.2649556822476361E+04 |
| 10 | 0.3274265467688794E+02 | 0.2556600916018281E+03 | -0.6721840706282554E+05 | 0.1518592350520692E+06 |
| 11 | 0.6156246406139704E+02 | 0.8689415173941595E+03 | -0.4582781585774253E+06 | 0.1681219443777066E+07 |
| 12 | 0.6100251528818768E+02 | 0.9210714604234481E+03 | -0.4398108606697874E+06 | 0.2110501110881329E+07 |
| 13 | 0.2812765033735517E+02 | 0.2028099038820981E+03 | -0.4176513439780750E+05 | 0.1123404978932170E+06 |
| 14 | 0.1015812424255807E+02 | 0.2737735240302546E+02 | -0.1830285725010282E+04 | 0.2059205431509025E+04 |
| 15 | 0.6628403769520118E+01 | 0.1163426581120765E+02 | -0.4778087399330668E+03 | 0.3921429848253429E+03 |
| 16 | 0.5078543964081808E+01 | 0.6821526275457762E+01 | -0.2010092890344462E+03 | 0.1416395150637668E+03 |
| 17 | 0.4157230014381144E+01 | 0.4647332376945644E+01 | -0.1029639727889250E+03 | 0.6900108261574451E+02 |
| 18 | 0.3542750915854995E+01 | 0.3461262937976936E+01 | -0.5935416050628737E+02 | 0.3992858476875006E+02 |
| 19 | 0.3103594576952411E+01 | 0.2741122158514052E+01 | -0.3705758280624042E+02 | 0.2593028736908622E+02 |
| 20 | 0.2774726489396073E+01 | 0.2272700955298311E+01 | -0.2450838813526847E+02 | 0.1831531891184402E+02 |
| 21 | 0.2520030316711311E+01 | 0.1953542956434740E+01 | -0.1692610534720381E+02 | 0.1379948511978922E+02 |
| 22 | 0.2317746477011871E+01 | 0.1729435681738751E+01 | -0.1208678115494804E+02 | 0.1094947853779263E+02 |
| 23 | 0.2153989967659938E+01 | 0.1569343871400643E+01 | -0.8860737242201525E+01 | 0.9067922695257138E+01 |
| 24 | 0.2019613499068920E+01 | 0.1453979589914899E+01 | -0.6634305010569030E+01 | 0.7782303804931697E+01 |
| 25 | 0.1908689264383096E+01 | 0.1369702175090221E+01 | -0.5057161812398873E+01 | 0.6871911971119896E+01 |
| 26 | 0.1816480620725851E+01 | 0.1309778919921657E+01 | -0.3914078123461568E+01 | 0.6227684238310211E+01 |
| 27 | 0.1739579447958521E+01 | 0.1269582830244330E+01 | -0.3071345782683535E+01 | 0.5780159160620986E+01 |
| 28 | 0.1675806662637884E+01 | 0.1244888298908247E+01 | -0.2444977856656303E+01 | 0.5477247049124202E+01 |
| 29 | 0.1623715877942706E+01 | 0.1232393728379181E+01 | -0.1980397742371957E+01 | 0.5283799338809050E+01 |
| 30 | 0.1582439224057063E+01 | 0.1229375405035280E+01 | -0.1642043586278724E+01 | 0.5175383378260868E+01 |
| 31 | 0.1551645793430460E+01 | 0.1233261198408628E+01 | -0.1407564173157425E+01 | 0.5133272082149876E+01 |
| 32 | 0.1531605663858521E+01 | 0.1241058308193479E+01 | -0.1264836121919516E+01 | 0.5139553554635057E+01 |
| 33 | 0.1523648147115839E+01 | 0.1247661080839741E+01 | -0.1212277766568493E+01 | 0.5165584746963301E+01 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.2883744276236732E+03 | 0.1226826052773576E+04 | 0.6417842770899591E+04 | 0.3825605145780512E+05 |
| 2 | 0.2945575697221625E+03 | 0.1259983767686758E+04 | 0.6636425081498122E+04 | 0.3985498527928310E+05 |
| 3 | 0.3112787561988349E+03 | 0.1366297471885140E+04 | 0.7359858586642650E+04 | 0.4522323418839895E+05 |
| 4 | 0.3392105668005165E+03 | 0.1569530839751625E+04 | 0.8827379109369643E+04 | 0.5651381661364262E+05 |
| 5 | 0.3814330994522609E+03 | 0.1923072395301514E+04 | 0.1161646934232287E+05 | 0.7938935578380853E+05 |
| 6 | 0.4448304769119560E+03 | 0.2545977097009550E+04 | 0.1717311161049583E+05 | 0.1297540490133080E+06 |
| 7 | 0.5446496163994199E+03 | 0.3737043804325109E+04 | 0.2974640621146022E+05 | 0.2620703074736724E+06 |
| 8 | 0.7211506188528558E+03 | 0.6456629931196199E+04 | 0.6620851597983128E+05 | 0.7433086159194873E+06 |
| 9 | 0.1182998791891019E+04 | 0.1732608726269338E+05 | 0.2893233439035744E+06 | 0.5282835308584735E+07 |
| 10 | 0.3274265467688794E+04 | 0.1325184925975741E+06 | 0.6080764730931759E+07 | 0.3039713292196019E+09 |
| 11 | 0.6156246406139704E+04 | 0.4650189083531020E+06 | 0.3929732624606907E+08 | 0.3579815433343165E+10 |
| 12 | 0.6100251528818768E+04 | 0.4633167617304785E+06 | 0.4006511608093550E+08 | 0.3796915891006544E+10 |
| 13 | 0.2812765033735517E+04 | 0.9919465183437936E+05 | 0.4035591063918472E+07 | 0.1821874585616623E+09 |
| 14 | 0.1015812424255807E+04 | 0.1302910670062412E+05 | 0.1933877837769796E+06 | 0.3191401635476486E+07 |
| 15 | 0.6628403769520118E+03 | 0.5545365968488408E+04 | 0.5370275276702704E+05 | 0.5803346683749134E+06 |
| 16 | 0.5078543964081808E+03 | 0.3254491980781495E+04 | 0.2432499115298454E+05 | 0.2042367207928665E+06 |
| 17 | 0.4157230014381144E+03 | 0.2188342044564764E+04 | 0.1353941413080371E+05 | 0.9473108755628928E+05 |
| 18 | 0.3542750915854995E+03 | 0.1597773436038857E+04 | 0.8535907050579309E+04 | 0.5189140967512707E+05 |
| 19 | 0.3103594576952411E+03 | 0.1234601023501733E+04 | 0.5862884983477898E+04 | 0.3185917839479603E+05 |
| 20 | 0.2774726489396073E+03 | 0.9949081036701582E+03 | 0.4291133377587108E+04 | 0.2128186848497809E+05 |
| 21 | 0.2520030316711311E+03 | 0.8284560324014502E+03 | 0.3300481858992863E+04 | 0.1518113949517662E+05 |
| 22 | 0.2317746477011871E+03 | 0.7084090056622305E+03 | 0.2642669569636361E+04 | 0.1141922898935442E+05 |
| 23 | 0.2153989967659938E+03 | 0.6193323213466298E+03 | 0.2188098540108460E+04 | 0.8976334656263684E+04 |
| 24 | 0.2019613499068920E+03 | 0.5518278479637157E+03 | 0.1864238674547831E+04 | 0.7324573613436629E+04 |
| 25 | 0.1908689264383096E+03 | 0.4999099861310605E+03 | 0.1628183348224629E+04 | 0.6172300671684809E+04 |
| 26 | 0.1816480620725851E+03 | 0.4596282976195013E+03 | 0.1453389504648442E+04 | 0.5349653329325441E+04 |
| 27 | 0.1739579447958521E+03 | 0.4283023657701561E+03 | 0.1322836069597042E+04 | 0.4753371659784834E+04 |
| 28 | 0.1675806662637884E+03 | 0.4040767386460686E+03 | 0.1225300092746271E+04 | 0.4318520694260544E+04 |
| 29 | 0.1623715877942706E+03 | 0.3856523043378643E+03 | 0.1153238885061903E+04 | 0.4003249182134647E+04 |
| 30 | 0.1582439224057063E+03 | 0.3721195548819248E+03 | 0.1101542600856702E+04 | 0.3780232641871605E+04 |
| 31 | 0.1551645793430460E+03 | 0.3628533254694983E+03 | 0.1066782157321171E+04 | 0.3631723305687628E+04 |
| 32 | 0.1531605663858521E+03 | 0.3574463634675046E+03 | 0.1046753830332733E+04 | 0.3546647223318597E+04 |
| 33 | 0.1523648147115839E+03 | 0.3556688146240874E+03 | 0.1040213879267877E+04 | 0.3518933305317931E+04 |

```
MODULATION TECHNIQUE = QPRS
QPRS CLASS 3 FILTER
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =            2
BIT RATE =  0.2400000000000000E+04
CARRIER FREQUENCY =       2400 HZ
MAXIMUM CARRIER AMPLITUDE =            1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.6784342273548912E-04 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES          SUM OF VARIANCES**2
 0.2212313780321216E+04   0.9349388767869366E+06
   SUM OF SKEWNESS             SUM OF SKEWNESS**2
-0.7876831322230185E+06   0.1444178694786092E+12
   SUM OF KURTOSIS             SUM OF KURTOSIS**2
 0.2375351460755018E+07   0.1504008831530188E+13
    MEAN VARIANCE          VARIANCE OF THE VARIANCES
 0.6703981152438534E+02   0.2458205556000923E+05
    MEAN SKEWNESS          VARIANCE OF THE SKEWNESS
-0.2386918582493996E+05   0.3925516075762725E+10
    MEAN KURTOSIS          VARIANCE OF THE KURTOSIS
 0.7198034729560660E+05   0.4165719401362236E+11
```

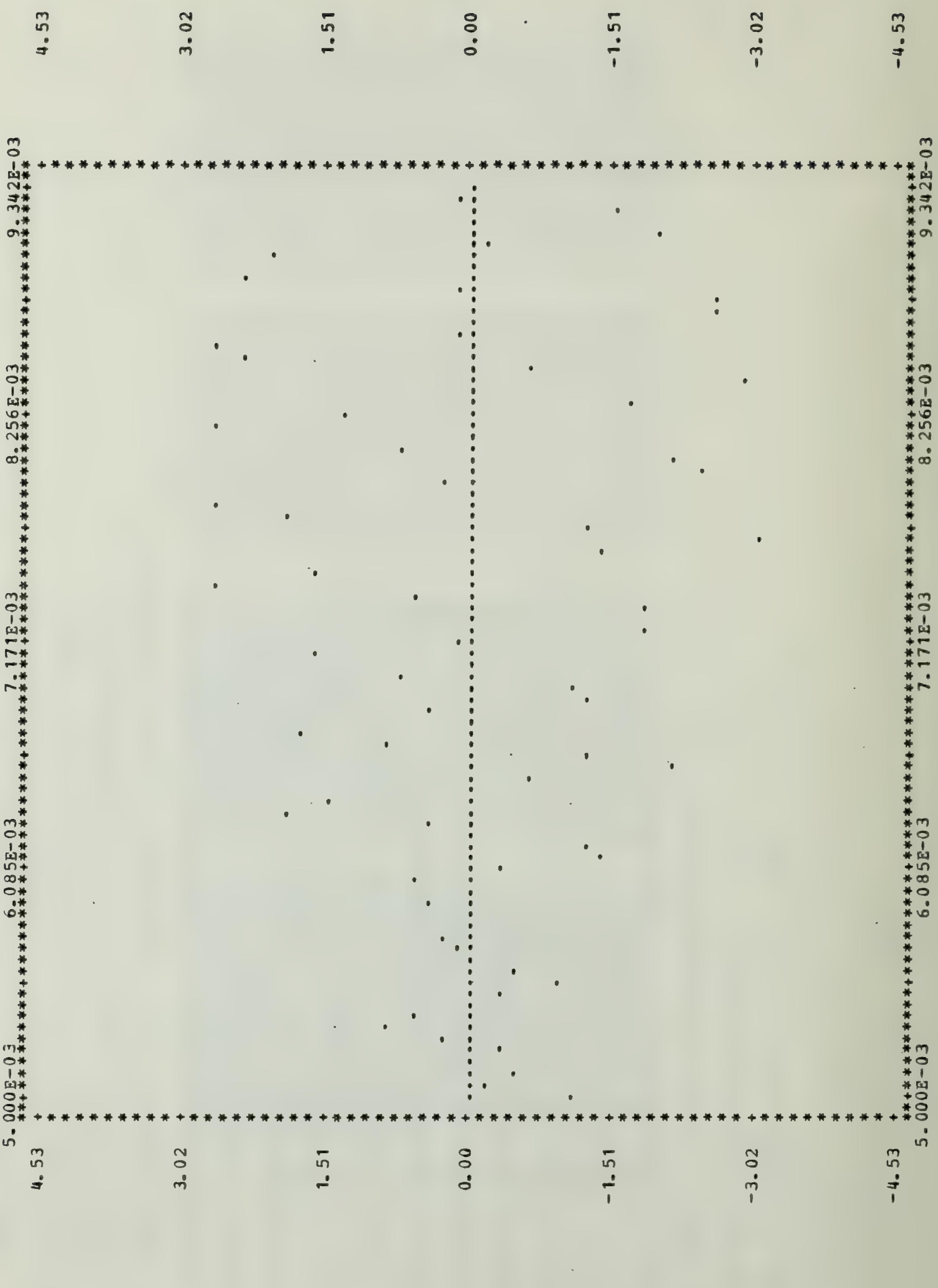

7.83
5.22
2.61
0.00
-2.61
-5.22
-7.83
5.000E-03
6.085E-03
7.171E-03
8.256E-03
9.342E-03

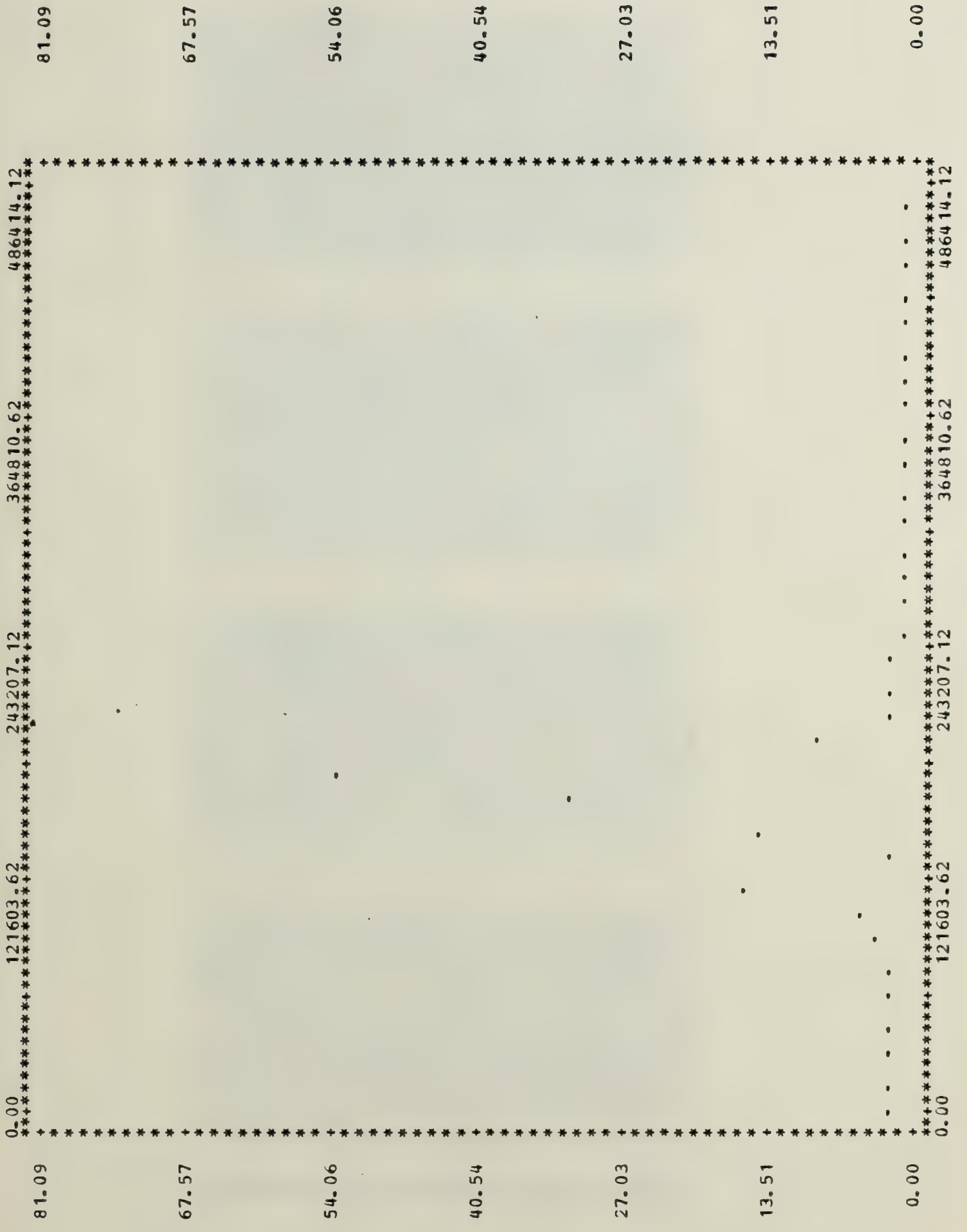

0.00
121603.62
243207.12
364810.62
486414.12
134.35
111.96
89.57
67.18
44.78
22.39
0.00

| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.2859193411467364E+01 | 0.4053038110163085E+01 | -0.2388872277111327E+02 | 0.5315861712866549E+02 |
| 2 | 0.2941193920455125E+01 | 0.3960147196244054E+01 | -0.2682550486427233E+02 | 0.5214126306947926E+02 |
| 3 | 0.3144223362768181E+01 | 0.3965144755612565E+01 | -0.3529891766507135E+02 | 0.5435317557178752E+02 |
| 4 | 0.3472939022742847E+01 | 0.4207631276200674E+01 | -0.5205508235858120E+02 | 0.6262547090742004E+02 |
| 5 | 0.3969306386409815E+01 | 0.4860943786245437E+01 | -0.8500175932943047E+02 | 0.8273400379153577E+02 |
| 6 | 0.4735423482398556E+01 | 0.6300931752245403E+01 | -0.1566149435795907E+03 | 0.1321350345761201E+03 |
| 7 | 0.6018537756096500E+01 | 0.9736414158625507E+01 | -0.3457034404393740E+03 | 0.2903044306984410E+03 |
| 8 | 0.8654258686250665E+01 | 0.2089270222273523E+02 | -0.1095988817483587E+04 | 0.1229022976292699E+04 |
| 9 | 0.2281137743621147E+02 | 0.1439177760640681E+03 | -0.2193236345074323E+05 | 0.5403321656593276E+05 |
| 10 | 0.4754546111151572E+02 | 0.4646395551694064E+03 | -0.2077441296070791E+06 | 0.5503560665874106E+06 |
| 11 | 0.4244750014125263E+02 | 0.3748277455711410E+03 | -0.1463406628119970E+06 | 0.4240689354081044E+06 |
| 12 | 0.4734966449442010E+02 | 0.4555200840149120E+03 | -0.2070845754790160E+06 | 0.4079376089497462E+06 |
| 13 | 0.4636502495407004E+02 | 0.5858023419797556E+03 | -0.1907034976555182E+06 | 0.9229169440607270E+06 |
| 14 | 0.1775571715328273E+02 | 0.7836469157643520E+02 | -0.1046518127146919E+05 | 0.1312130526653664E+05 |
| 15 | 0.8010362291212992E+01 | 0.1596765263482768E+02 | -0.8825082674162869E+03 | 0.6106026606095356E+03 |
| 16 | 0.5748360868725290E+01 | 0.7781191394513882E+01 | -0.3051632597123039E+03 | 0.1547481205147622E+03 |
| 17 | 0.4587361831012248E+01 | 0.4764166354630095E+01 | -0.1458247303643428E+03 | 0.6122038538089117E+02 |
| 18 | 0.3859990888844249E+01 | 0.3291449429627744E+01 | -0.8182688020467373E+02 | 0.3123619875679458E+02 |
| 19 | 0.3356671073359412E+01 | 0.2458185419647806E+01 | -0.5067489115592908E+02 | 0.1885213610174446E+02 |
| 20 | 0.2985704246868344E+01 | 0.1947285393959427E+01 | -0.3354116060432313E+02 | 0.1288351792221858E+02 |
| 21 | 0.2700289329982362E+01 | 0.1617844173346278E+01 | -0.2327974567069668E+02 | 0.9657495559635674E+01 |
| 22 | 0.2474887178278520E+01 | 0.1393588316418582E+01 | -0.1674673133820691E+02 | 0.7685551297040409E+01 |
| 23 | 0.2293436153883331E+01 | 0.1235603072617695E+01 | -0.1238328622708189E+02 | 0.6375226655986307E+01 |
| 24 | 0.2143603457021659E+01 | 0.1129741939228789E+01 | -0.9340535778002867E+01 | 0.5530298158106696E+01 |
| 25 | 0.2017352208785796E+01 | 0.1065020962976242E+01 | -0.7145984024478282E+01 | 0.5008270937598276E+01 |
| 26 | 0.1909802109242861E+01 | 0.1031153159062995E+01 | -0.5524790323883738E+01 | 0.4698582156733970E+01 |
| 27 | 0.1817803178471110E+01 | 0.1020234834778769E+01 | -0.4307541772447900E+01 | 0.4531591295706342E+01 |
| 28 | 0.1739297395622849E+01 | 0.1026428800839872E+01 | -0.3385787401612433E+01 | 0.4465390207474467E+01 |
| 29 | 0.1673014887490447E+01 | 0.1045320201357048E+01 | -0.2688643953287969E+01 | 0.4474563821840981E+01 |
| 30 | 0.1618309679485713E+01 | 0.1073390686558981E+01 | -0.2169671349142156E+01 | 0.4543413084630725E+01 |
| 31 | 0.1575127269274742E+01 | 0.1107432254242980E+01 | -0.1799539780561719E+01 | 0.4660898333893368E+01 |
| 32 | 0.1544328394719743E+01 | 0.1143084440129354E+01 | -0.1563241471185823E+01 | 0.4811604062719301E+01 |
| 33 | 0.1530498278895295E+01 | 0.1165859218662538E+01 | -0.1470070156508873E+01 | 0.4923039669179051E+01 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.2859193411467364E+03 | 0.1218749469323984E+04 | 0.6430051688814439E+04 | 0.3912446482248652E+05 |
| 2 | 0.2941193920455125E+03 | 0.1257116740200380E+04 | 0.6679597518767498E+04 | 0.4099902143818628E+05 |
| 3 | 0.3144223362768182E+03 | 0.1381163386303369E+04 | 0.7520938683542618E+04 | 0.4741984235567450E+05 |
| 4 | 0.3472939022742847E+03 | 0.1622686041912871E+04 | 0.9288699702861901E+04 | 0.6151115249264073E+05 |
| 5 | 0.3969306386409816E+03 | 0.2056772753757673E+04 | 0.1284062910987031E+05 | 0.9218510865065588E+05 |
| 6 | 0.4735423482398556E+03 | 0.2866215799237462E+04 | 0.2057189533450305E+05 | 0.1680977610919490E+06 |
| 7 | 0.6018537756096500E+03 | 0.4586184673859834E+04 | 0.4099147345992035E+05 | 0.4127460555429674E+06 |
| 8 | 0.8654258686250665E+03 | 0.9557996860905298E+04 | 0.1235740116398711E+06 | 0.1788354599317327E+07 |
| 9 | 0.2281137743621147E+04 | 0.6628375388407253E+05 | 0.2238763050033797E+07 | 0.8396436177827680E+08 |
| 10 | 0.4754546111151572E+04 | 0.2720564031924366E+06 | 0.1757861427920527E+08 | 0.1241206409612254E+10 |
| 11 | 0.4244750014125263E+04 | 0.2172869736357072E+06 | 0.1267544219743408E+08 | 0.8194713555490152E+09 |
| 12 | 0.4734966449442010E+04 | 0.2692955610908911E+06 | 0.1709630770902248E+08 | 0.1164219318416618E+10 |
| 13 | 0.4636502495407004E+04 | 0.2729659857551496E+06 | 0.1846793135009938E+08 | 0.1382943828406290E+10 |
| 14 | 0.1775571715328273E+04 | 0.3928465362880496E+05 | 0.9830103694208338E+06 | 0.2663547206451179E+08 |
| 15 | 0.8010362291212992E+03 | 0.7997388014496647E+04 | 0.9110191858964195E+05 | 0.1136318940332887E+07 |
| 16 | 0.5748360868725290E+03 | 0.4074703215766091E+04 | 0.3314353704488787E+05 | 0.2972652853806565E+06 |
| 17 | 0.4587361831012248E+03 | 0.2576041325971184E+04 | 0.1666045021145127E+05 | 0.1194261062431947E+06 |
| 18 | 0.3859990888844249E+03 | 0.1815806459729208E+04 | 0.9864395218518330E+04 | 0.5970034454704064E+05 |
| 19 | 0.3356671073359412E+03 | 0.1370084426017916E+04 | 0.6475831028085240E+04 | 0.3429694022355070E+05 |
| 20 | 0.2985704246868344E+03 | 0.1084224238978750E+04 | 0.4574516714243508E+04 | 0.2176964006173594E+05 |
| 21 | 0.2700289329982362E+03 | 0.8893228197229408E+03 | 0.3417999702830066E+04 | 0.1492684967005119E+05 |
| 22 | 0.2474887178278520E+03 | 0.7504718978461813E+03 | 0.2672313390051410E+04 | 0.1089804187849496E+05 |
| 23 | 0.2293436153883331E+03 | 0.6483096433830683E+03 | 0.2170271824041791E+04 | 0.8386780312190165E+04 |
| 24 | 0.2143603457021659E+03 | 0.5713480300791711E+03 | 0.1821170103329349E+04 | 0.6750614232021021E+04 |
| 25 | 0.2017352208785796E+03 | 0.5124080687639409E+03 | 0.1572582258655386E+04 | 0.5647284703365001E+04 |
| 26 | 0.1909802109242862E+03 | 0.4668185723940848E+03 | 0.1392645430893823E+04 | 0.4883614843619811E+04 |
| 27 | 0.1817803178471110E+03 | 0.4314440882090652E+03 | 0.1261205448260302E+04 | 0.4345381869358503E+04 |
| 28 | 0.1739297395622849E+03 | 0.4041319943251899E+03 | 0.1165107349407684E+04 | 0.3962508764554495E+04 |
| 29 | 0.1673014887490447E+03 | 0.3833845813108151E+03 | 0.1095564178420125E+04 | 0.3690809901269517E+04 |
| 30 | 0.1618309679485713E+03 | 0.3681582998410544E+03 | 0.1046630041989700E+04 | 0.3501983236805778E+04 |
| 31 | 0.1575127269274742E+03 | 0.3577383846113457E+03 | 0.1014291293664126E+04 | 0.3377938995533378E+04 |
| 32 | 0.1544328394719743E+03 | 0.3516603786465719E+03 | 0.9959231388592782E+03 | 0.3307531236388883E+04 |
| 33 | 0.1530498278895295E+03 | 0.3496625608177372E+03 | 0.9899818305858252E+03 | 0.3284702451033463E+04 |

```
MODULATION TECHNIQUE = QPRS
QPRS CLASS 4 FILTER
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           2
BIT RATE =  0.2400000000000000E+04
CARRIER FREQUENCY =       2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =          0 DEGREES
TIME BETWEEN SAMPLES =  0.6784342273548912E-04 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =          1
NUMBER OF TIMES SIMULATION REPEATS =        100
   SUM OF VARIANCES          SUM OF VARIANCES**2
 0.7516311188359808E+03   0.1147433777377834E+06
   SUM OF SKEWNESS            SUM OF SKEWNESS**2
-0.1502958760239533E+06   0.5926297375617011E+10
   SUM OF KURTOSIS            SUM OF KURTOSIS**2
 0.3178921864476514E+06   0.3285296126177793E+11
    MEAN VARIANCE          VARIANCE OF THE VARIANCES
 0.2277670057078729E+02   0.3050740650136574E+04
    MEAN SKEWNESS          VARIANCE OF THE SKEWNESS
-0.4554420485574342E+04   0.1638058362173805E+09
    MEAN KURTOSIS          VARIANCE OF THE KURTOSIS
 0.9633096559019739E+04   0.9309585979490561E+09
```

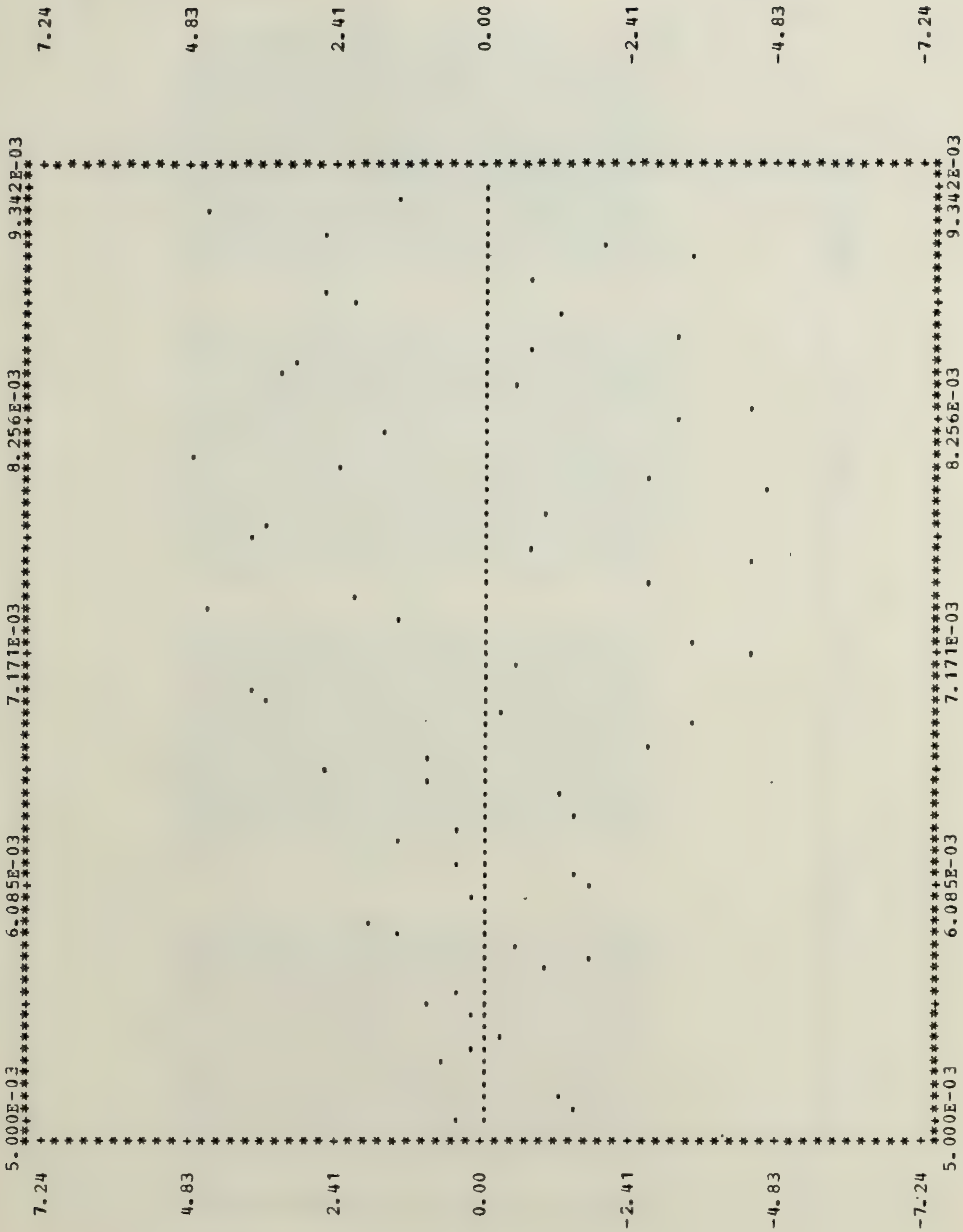

4.53
3.02
1.51
0.00
-1.51
-3.02
-4.53
5.000E-03
6.085E-03
7.171E-03
8.256E-03
9.342E-03

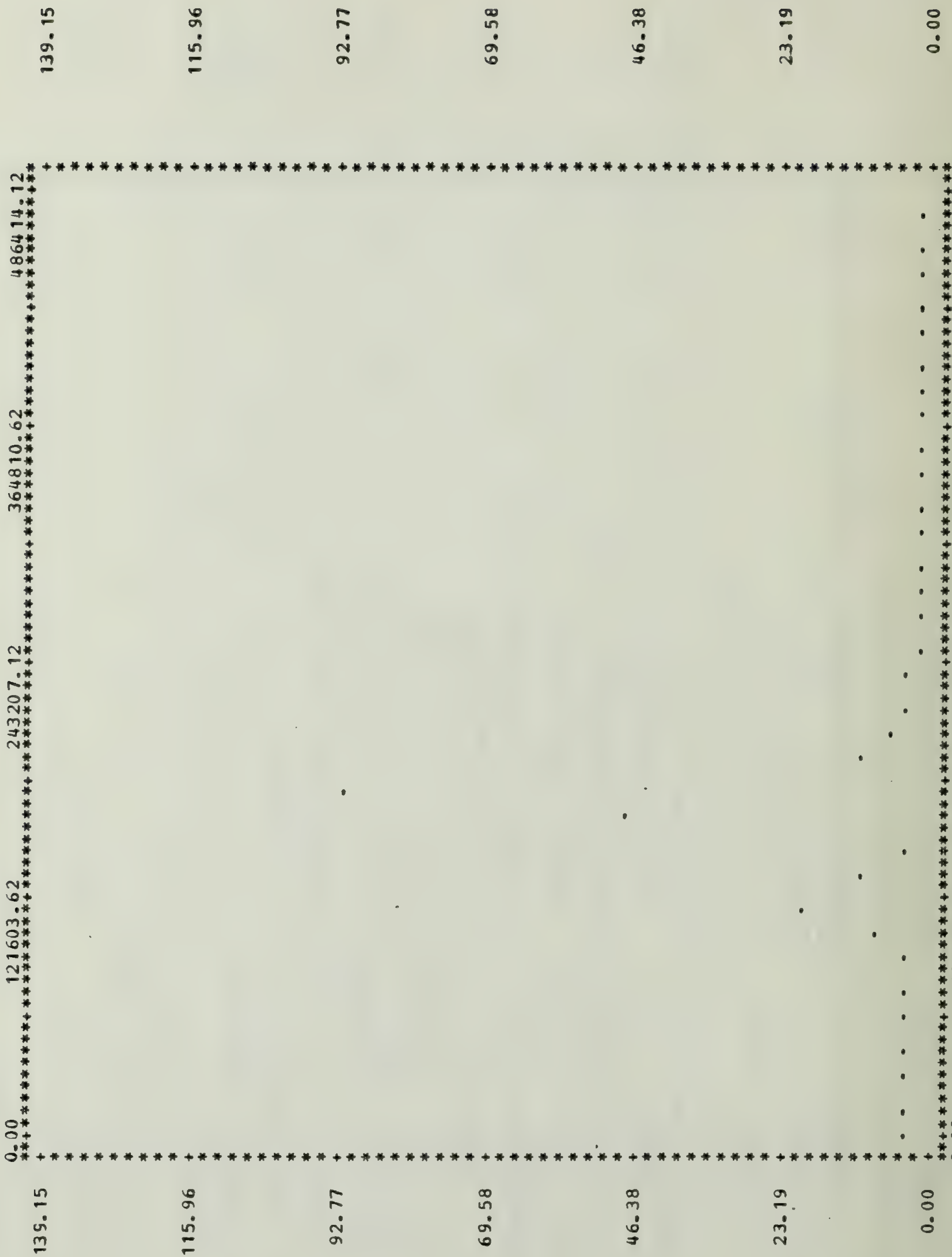

0.00
121603.62
243207.12
364810.62
486414.12
81.09
67.57
54.06
40.54
27.03
13.51
0.00

| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.1639783109841063E+01 | 0.1709248720151234E+01 | -0.1134099002226032E+01 | 0.1062328648097734E+02 |
| 2 | 0.1694119010472507E+01 | 0.1659799398022893E+01 | -0.1631841391220386E+01 | 0.1036162216583215E+02 |
| 3 | 0.1822080267905823E+01 | 0.1637975366559496E+01 | -0.3033602018908517E+01 | 0.1051811011425401E+02 |
| 4 | 0.2024944483212605E+01 | 0.1696003179844438E+01 | -0.5852717582052459E+01 | 0.1149726415338565E+02 |
| 5 | 0.2329575301573474E+01 | 0.1889410056086123E+01 | -0.1162556661054461E+02 | 0.1396880835853768E+02 |
| 6 | 0.2799618459186339E+01 | 0.2351081202562522E+01 | -0.2485697728770199E+02 | 0.2003885906538744E+02 |
| 7 | 0.3593606690701501E+01 | 0.3517560421894356E+01 | -0.6221532265140116E+02 | 0.3923065739628220E+02 |
| 8 | 0.5250268678825291E+01 | 0.7679872826672729E+01 | -0.2240112630638912E+03 | 0.1615596553464001E+03 |
| 9 | 0.1468732445728772E+02 | 0.5698432351652923E+02 | -0.5766331711517894E+04 | 0.7700144626832553E+04 |
| 10 | 0.2971531933822615E+02 | 0.1977041525414051E+03 | -0.4980046600273765E+05 | 0.1015887560620818E+06 |
| 11 | 0.1946196225072521E+02 | 0.8520733202076775E+02 | -0.1338196665730689E+05 | 0.3082167317768066E+05 |
| 12 | 0.2493319558C19601E+02 | 0.1203157758015537E+03 | -0.2953935348818128E+05 | 0.3229216284253984E+05 |
| 13 | 0.2955681866542102E+02 | 0.2229606796898662E+03 | -0.4852116398403028E+05 | 0.1430962797654248E+06 |
| 14 | 0.1134843594184110E+02 | 0.2883361550590456E+02 | -0.2628615664034768E+04 | 0.2014017364065328E+04 |
| 15 | 0.4970090484803092E+01 | 0.5007521697074969E+01 | -0.1926957672498994E+03 | 0.6163959197677531E+02 |
| 16 | 0.3532757103156359E+01 | 0.2436613452379979E+01 | -0.6195368080461159E+02 | 0.1530304581177205E+02 |
| 17 | 0.2812436795395693E+01 | 0.1543256466260907E+01 | -0.2803994388113845E+02 | 0.6634640987626415E+01 |
| 18 | 0.2369246425223995E+01 | 0.1116129068281124E+01 | -0.1499345615149528E+02 | 0.3767765889897674E+01 |
| 19 | 0.2067752681199091E+01 | 0.8730631986165904E+00 | -0.8863085844789377E+01 | 0.2483963240368716E+01 |
| 20 | 0.1852664563048547E+01 | 0.7086180491037625E+00 | -0.5611531249491262E+01 | 0.1721796443868016E+01 |
| 21 | 0.1691049257918267E+01 | 0.5969980314448385E+00 | -0.3716680154916490E+01 | 0.1275128193869637E+01 |
| 22 | 0.1564485152670945E+01 | 0.5228065905688973E+00 | -0.2532982447521070E+01 | 0.1015577885269764E+01 |
| 23 | 0.1462724537501172E+01 | 0.4737758456720625E+00 | -0.1755262251553210E+01 | 0.8592948826267097E+00 |
| 24 | 0.1379361198878535E+01 | 0.4420632119864455E+00 | -0.1224165844748508E+01 | 0.7627903599458743E+00 |
| 25 | 0.1310212971454939E+01 | 0.4225151882274944E+00 | -0.8506267979152597E+00 | 0.7026301237469474E+00 |
| 26 | 0.1252466606083759E+01 | 0.4116512857478918E+0C | -0.5821187634403218E+00 | 0.6656761386153749E+00 |
| 27 | 0.1204196957391062E+01 | 0.4070568429211903E+00 | -0.3863199563233570E+00 | 0.6442916119914803E+00 |
| 28 | 0.1164070364847772E+01 | 0.4070406361202367E+00 | -0.2426723420547243E+00 | 0.6339971166406047E+00 |
| 29 | 0.1131163935402371E+01 | 0.4104109765706129E+00 | -0.1377979566892886E+00 | 0.6322034788189101E+00 |
| 30 | 0.1104905018984923E+01 | 0.4162098554263231E+00 | -0.6292504569035784E-01 | 0.6371923425461106E+00 |
| 31 | 0.1085094957871504E+01 | 0.4234143765176991E+00 | -0.1239972092372565E-01 | 0.6472465033331667E+00 |
| 32 | 0.1072017799482293E+01 | 0.4305494285120794E+00 | 0.1715449376312383E-01 | 0.6596579907407305E+00 |
| 33 | 0.1066896145745031E+01 | 0.4345943867280897E+00 | 0.2713543267703411E-01 | 0.66785496309SC203E+00 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.1639783109841063E+03 | 0.4381044880269750E+03 | 0.1504001389749991E+04 | 0.6028224494587657E+04 |
| 2 | 0.1694119010472507E+03 | 0.4513240625687009E+03 | 0.1556598039393050E+04 | 0.6283642510452042E+04 |
| 3 | 0.1822080267905823E+03 | 0.4941572115585657E+03 | 0.1733826810843475E+04 | 0.7151628698908221E+04 |
| 4 | 0.2024944483212605E+03 | 0.5779443308139159E+03 | 0.2105563762633040E+04 | 0.9029491598424876E+04 |
| 5 | 0.2329575301573474E+03 | 0.7297437041226407E+03 | 0.2852037850613845E+04 | 0.1304690135038802E+05 |
| 6 | 0.2799618459186339E+03 | 0.1016543390755379E+04 | 0.4484530491794313E+04 | 0.2284824450720208E+05 |
| 7 | 0.3593606690701501E+03 | 0.1639639386513001E+04 | 0.8872323134501433E+04 | 0.5444320652383947E+05 |
| 8 | 0.5250268678825291E+03 | 0.3516839529825987E+04 | 0.2747886674934024E+05 | 0.2395389115966948E+06 |
| 9 | 0.1468732445728772E+04 | 0.2721319799950061E+05 | 0.5792905345227176E+06 | 0.1354096031979063E+08 |
| 10 | 0.2971531933822615E+04 | 0.1078727314388748E+06 | 0.4459771346830303E+07 | 0.2026499395458332E+09 |
| 11 | 0.1946196225072521E+04 | 0.4631232333492131E+05 | 0.1290035804802437E+07 | 0.4129848397914642E+08 |
| 12 | 0.2493319558019601E+04 | 0.7407768598838439E+05 | 0.2462712101444010E+07 | 0.8847358410767536E+08 |
| 13 | 0.2955681866542102E+04 | 0.1094336602513548E+06 | 0.4676695051500612E+07 | 0.2225674204211198E+09 |
| 14 | 0.1134843594184110E+04 | 0.1573322776769165E+05 | 0.2470236163714422E+06 | 0.4233144694722940E+07 |
| 15 | 0.4970090484803092E+03 | 0.2965924590723445E+04 | 0.2001280415057830E+05 | 0.1474973714749340E+06 |
| 16 | 0.3532757103156359E+03 | 0.1489262006775789E+04 | 0.7092160168252687E+04 | 0.3695846281762211E+05 |
| 17 | 0.2812436795395693E+03 | 0.9437624629693893E+03 | 0.3576862297172163E+04 | 0.1488190681227194E+05 |
| 18 | 0.2369246425223995E+03 | 0.6718296401034992E+03 | 0.2153178579687351E+04 | 0.7608053377373551E+04 |
| 19 | 0.2067752681199091E+03 | 0.5139933717236453E+03 | 0.1447004702900242E+04 | 0.4515033391001683E+04 |
| 20 | 0.1852664563048547E+03 | 0.4133897851788587E+03 | 0.1049991495597069E+04 | 0.2974218440064030E+04 |
| 21 | 0.1691049257918267E+03 | 0.3450675643836312E+03 | 0.8069812120149764E+03 | 0.2118738265791955E+04 |
| 22 | 0.1564485152670945E+03 | 0.2965192317591039E+03 | 0.6488788033678661E+03 | 0.1604859418202128E+04 |
| 23 | 0.1462724537501172E+03 | 0.2608601159823359E+03 | 0.5412606378313685E+03 | 0.1277349367688637E+04 |
| 24 | 0.1379361198878536E+03 | 0.2340279896838212E+03 | 0.4654833373950505E+03 | 0.1058943962953624E+04 |
| 25 | 0.1310212971454939E+03 | 0.2134948066914001E+03 | 0.4107767092907294E+03 | 0.9081718957376042E+03 |
| 26 | 0.1252466606083759E+03 | 0.1976207372245383E+03 | 0.3705937259151374E+03 | 0.8013993822923571E+03 |
| 27 | 0.1204196957391062E+03 | 0.1853076586681869E+03 | 0.3407906981781624E+03 | 0.7244971790085427E+03 |
| 28 | 0.1164070364847772E+03 | 0.1758030044075860E+03 | 0.3186620053773817E+03 | 0.6686943101747503E+03 |
| 29 | 0.1131163935402371E+03 | 0.1685838715559886E+03 | 0.3024018213639951E+03 | 0.6283944137293207E+03 |
| 30 | 0.1104905018984923E+03 | 0.1632862857850134E+03 | 0.2907919853211530E+03 | 0.5999720029213764E+03 |
| 31 | 0.1085094957871504E+03 | 0.1596611300350684E+03 | 0.2830162758953530E+03 | 0.5810861131486764E+03 |
| 32 | 0.1072017799482293E+03 | 0.1575466096633816E+03 | 0.2785493263166431E+03 | 0.5702822834018495E+03 |
| 33 | 0.1066896145745031E+03 | 0.1568515828666411E+03 | 0.2770931137498631E+03 | 0.5667653419404973E+03 |

```
MODULATION TECHNIQUE = QPRS
QPRS CLASS 5 FILTER
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =          1200 HZ
BITS PER BINARY CODE WORD =            2
BIT RATE =  0.240000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =            1 VOLT(S)
INITIAL PHASE ANGLE =            0 DEGREES
TIME BETWEEN SAMPLES =  0.678434227354891 2E-04 SEC
NUMBER OF SAMPLES GENERATED =           64
SEED FOR RANDOM NUMBER GENERATOR =            1
NUMBER OF TIMES SIMULATION REPEATS =          100
   SUM OF VARIANCES          SUM OF VARIANCES**2
 0.242217786507317 6E+04   0.134853836921845 7E+07
   SUM OF SKEWNESS            SUM OF SKEWNESS**2
-0.820216075504924 9E+06   0.204913894994457 8E+12
   SUM OF KURTOSIS            SUM OF KURTOSIS**2
 0.358270392726342 5E+07   0.478288944229002 5E+13
    MEAN VARIANCE          VARIANCE OF THE VARIANCES
 0.733993292446416 9E+02   0.365860043315896 2E+05
    MEAN SKEWNESS          VARIANCE OF THE SKEWNESS
-0.248550325910583 3E+05   0.576648117831478 0E+10
    MEAN KURTOSIS          VARIANCE OF THE KURTOSIS
 0.108566785674649 2E+06   0.137310212277596 6E+12
```


7.24
4.83
2.41
0.00
-2.41
-4.83
-7.24
5.000E-03
6.085E-03
7.171E-03
8.256E-03
9.342E-03



0.00
121603.62
243207.12
364810.62
486414.12
139.15
115.96
92.77
69.58
46.38
23.19
0.00

| N | MEAN | VARIANCE | SKEWNESS | KURTOSIS |
|---|---|---|---|---|
| 1 | 0.3175007969977690E+01 | 0.5114157186225571E+01 | -0.3647377245722682E+02 | 0.6390177902993634E+02 |
| 2 | C.3240619944765044E+01 | 0.5086102701788736E+01 | -0.3945722174718513E+02 | 0.6384581606339765E+02 |
| 3 | 0.3418492879921329E+01 | 0.5172192798243949E+01 | -0.4849017265821298E+02 | 0.6741248641994679E+02 |
| 4 | 0.3720050986691612E+01 | 0.5489382820690838E+01 | -0.6651291925211635E+02 | 0.7734383790401080E+02 |
| 5 | 0.4188073611507133E+01 | 0.6215059554191426E+01 | -0.1016999797030395E+03 | 0.9959832869899236E+02 |
| 6 | 0.4921214356587952E+01 | 0.7760498167346390E+01 | -0.1769255927986075E+03 | 0.1526183467386106E+03 |
| 7 | 0.6164670833203001E+01 | 0.1134896285448964E+02 | -0.3719888818360394E+03 | 0.3136503593616817E+03 |
| 8 | 0.8734938742447645E+01 | 0.2239794782070082E+02 | -0.1126894580418845E+04 | 0.1163666961030995E+04 |
| 9 | 0.2001458895916663E+02 | 0.1331245996682923E+03 | -0.1451737131188666E+05 | 0.4314997043974916E+05 |
| 10 | 0.5409232156107556E+02 | 0.7186407819046712E+03 | -0.3060942961486485E+06 | 0.1245883717653442E+07 |
| 11 | 0.3071091723336924E+02 | 0.2359866864671304E+03 | -0.5416809541898984E+05 | 0.1854316904737086E+06 |
| 12 | 0.4125066118058745E+02 | 0.4007672219572611E+03 | -0.1350387560075282E+06 | 0.3465758238469681E+06 |
| 13 | 0.5405198483591909E+02 | 0.7711366140096073E+03 | -0.2996435115512078E+06 | 0.1753359099945396E+07 |
| 14 | 0.1566444696486908E+02 | 0.4844857365619617E+02 | -0.7151724018508688E+04 | 0.5637426265450937E+04 |
| 15 | 0.8027426339157971E+01 | 0.1187759149364716E+02 | -0.8962200092443305E+03 | 0.3718418536813451E+03 |
| 16 | 0.5774108803249598E+01 | 0.5966604758607288E+01 | -0.3115922473316341E+03 | 0.1010149760039774E+03 |
| 17 | 0.4596310476428844E+01 | 0.3860312828819921E+01 | -0.1472328835908598E+03 | 0.4684905154496522E+02 |
| 18 | 0.3863417960017863E+01 | 0.2848711022275091E+01 | -0.8198224752509745E+02 | 0.2796050352952778E+02 |
| 19 | 0.3363155825534746E+01 | 0.2278343999412112E+01 | -0.5072245028464429E+02 | 0.1915731497924338E+02 |
| 20 | 0.3003261081837905E+01 | 0.1911527321194257E+01 | -0.3385313123379364E+02 | 0.1406866837863103E+02 |
| 21 | 0.2731904280991412E+01 | 0.1671831119124026E+01 | -0.2389082914534472E+02 | 0.1105673656228376E+02 |
| 22 | 0.2521234441845605E+01 | 0.1509330772274732E+01 | -0.1761502851110826E+02 | 0.9136822096088569E+01 |
| 23 | 0.2354205282986497E+01 | 0.1396522177812353E+01 | -0.1345982833386097E+02 | 0.7843945020122483E+01 |
| 24 | 0.2219851857423473E+01 | 0.1317021271177417E+01 | -0.1059986556386777E+02 | 0.6935658792154644E+01 |
| 25 | 0.2110603554002656E+01 | 0.1261545612229099E+01 | -0.8569458694423458E+01 | 0.6286691888697518E+01 |
| 26 | 0.2021075511112747E+01 | 0.1224594843570620E+01 | -0.7092619024801998E+01 | 0.5826585347360673E+01 |
| 27 | 0.1947570339122554E+01 | 0.1201959396455356E+01 | -0.6000660348007621E+01 | 0.5506110818704212E+01 |
| 28 | 0.1887571446576071E+01 | 0.1190243399752315E+01 | -0.5187278999299626E+01 | 0.5290200850572500E+01 |
| 29 | 0.1839396649648418E+01 | 0.1186660468906729E+01 | -0.4584006084000971E+01 | 0.5153959026705035E+01 |
| 30 | 0.1802007198312946E+01 | 0.1188759606911689E+01 | -0.4146598216959648E+01 | 0.5078752597685412E+01 |
| 31 | 0.1774907328415459E+01 | 0.1194148135719785E+01 | -0.3847289024627549E+01 | 0.5048951807450610E+01 |
| 32 | 0.1758112351977182E+01 | C.1200173229585994E+01 | -0.3670463427801961E+01 | 0.5048651820880867E+01 |
| 33 | 0.1752263344775810E+01 | 0.1203202048864805E+01 | -0.3611032699601022E+01 | 0.5055288718097053E+01 |

| N | SUM X | SUM X**2 | SUM X**3 | SUM X**4 |
|---|---|---|---|---|
| 1 | 0.31750079699776 90E+03 | 0.1514369122378517E+04 | 0.8760889731512676E+04 | 0.5654457975856909E+05 |
| 2 | 0.3240619944765044E+03 | 0.1553685930117985E+04 | 0.9058671139123197E+04 | 0.5899532160598735E+05 |
| 3 | 0.3418492879921329E+03 | 0.1680656444033433E+04 | 0.1004970028273070E+05 | 0.6728814818325367E+05 |
| 4 | 0.3720050986691612E+03 | 0.1927326833606917E+04 | 0.1209021447318121E+05 | C.8506168038807575E+05 |
| 5 | 0.4188073611507133E+03 | 0.2369286953405192E+04 | 0.1609025441754881E+05 | 0.1224603791529354E+06 |
| 6 | 0.4921214356587952E+03 | 0.3190124392916030E+04 | 0.2456162845660115E+05 | 0.2111552340434820E+06 |
| 7 | 0.6164676833203001E+03 | 0.4923871368377453E+04 | 0.4626269098775424E+05 | 0.4826786734449781E+06 |
| 8 | 0.8734938742447645E+03 | 0.9847312317680666E+04 | 0.1300977206093252E+06 | 0.1900367841479010E+07 |
| 9 | 0.2001458895916663E+04 | 0.5323771248760041E+05 | 0.1664738912267298E+07 | 0.5777448275164625E+08 |
| 10 | 0.5409232156107556E+04 | 0.3637433625952424E+06 | 0.2783254334038806E+08 | 0.2329269091647567E+10 |
| 11 | 0.3071091723336924E+04 | 0.1176787256917315E+06 | 0.5236623185179429E+07 | 0.2627551467752967E+09 |
| 12 | 0.4125066118058745E+04 | 0.2098376597573313E+06 | 0.1212362760641124E+08 | 0.7613583395908520E+09 |
| 13 | 0.5405198483591909E+04 | 0.3685042312571939E+06 | 0.2920648079159632E+08 | 0.2590993316517971E+10 |
| 14 | 0.1566444696486908E+04 | 0.2933389866348303E+05 | 0.6142505180670135E+06 | 0.1392715907516543E+08 |
| 15 | 0.8027426339157971E+03 | 0.7619838920931783E+04 | 0.8099317131153205E+05 | 0.9374736197964726E+06 |
| 16 | 0.5774108803249599E+03 | 0.3924727118278573E+04 | 0.3015811298449788E+05 | 0.2550094891853301E+06 |
| 17 | 0.4596310476428844E+03 | 0.2494777969626127E+04 | 0.1545181999623744E+05 | 0.1064344796474776E+06 |
| 18 | 0.3863417960017863E+03 | 0.1774622224584093E+04 | 0.9384897704847650E+04 | 0.5573475741090785E+05 |
| 19 | 0.3363155825534747E+03 | 0.1356637766624629E+04 | 0.6353344174062120E+04 | 0.3369716251661412E+05 |
| 20 | 0.3003261081837905E+03 | 0.1091198917366450E+04 | 0.4642237174695144E+04 | 0.2252718248550863E+05 |
| 21 | 0.2731904280991412E+03 | 0.9118413808431990E+03 | 0.3591447001097654E+04 | 0.1622980981279217E+05 |
| 22 | 0.2521234441845605E+03 | 0.7850860575300507E+03 | 0.2905330550912102E+04 | 0.1239278015753117E+05 |
| 23 | 0.2354205282986497E+03 | 0.6924839470475764E+03 | 0.2436308763293834E+04 | 0.9914096961588693E+04 |
| 24 | 0.2219851857423473E+03 | 0.6231593327372087E+03 | 0.2104429196574328E+04 | 0.8239831778016403E+04 |
| 25 | 0.2110603554002656E+03 | 0.5703577518275451E+03 | 0.1863521952304470E+04 | 0.7070002419209624E+04 |
| 26 | 0.2021075511112747E+03 | 0.5297095116754564E+03 | 0.1685537619348139E+04 | 0.6232225707108218E+04 |
| 27 | 0.1947570339122554E+03 | 0.4982970028320744E+03 | 0.1552733308401496E+04 | 0.5622630355034676E+04 |
| 28 | 0.1887571446576071E+03 | 0.4741266931684075E+03 | 0.1453530931136841E+04 | 0.5176256608601879E+04 |
| 29 | 0.1839396649648418E+03 | 0.4558173898955488E+03 | 0.1380210324128415E+04 | 0.4851370285803680E+04 |
| 30 | 0.1802007198312946E+03 | 0.4424101953614247E+03 | 0.1327573259694673E+04 | 0.4620766430978244E+04 |
| 31 | 0.1774907328415459E+03 | 0.4332502678825489E+03 | 0.1292149119152755E+04 | 0.4466784878181024E+04 |
| 32 | 0.1758112351977182E+03 | 0.4279130539464874E+03 | 0.1271719525931649E+04 | 0.4378406409209604E+04 |
| 33 | 0.1752263344775810E+03 | 0.4261596857821065E+03 | 0.1265043353402402E+04 | 0.4349588740060286E+04 |

# APPENDIX D

## FORTRAN PROGRAM F-TEST

```
C       ***********************************************************************
C       *                                                                     *
C       * THIS PROGRAM COMPUTES THE F STATISTIC FOR A GIVEN SET OF INPUT      *
C       * DATA.  THE DATA MUST BE FORMATTED AS E23.16 IN TWO COLUMNS          *
C       * STARTING IN THE SECOND COLUMN WITH TWO BLANK COLUMNS BETWEEN        *
C       * THE DATA.  THIS PROGRAM IS NOT MEANT TO BE EXTREMELY VERSATILE      *
C       * AND WAS DESIGNED TO DO THE ANALYSIS FOR SPECIFIC SETS OF DATA.      *
C       * THE INPUT DATA MUST BE THE SUM OF THE OBSERVATIONS FOR A SET        *
C       * OF N OBSERVATIONS AND THE SUM OF THE SQUARE OF THE OBSERVATIONS*
C       * FOR THE SET OF N OBSERVATIONS.  THE INDEX FOR THE DO LOOP MUST      *
C       * BE CHANGED TO ACCOUNT FOR THE NUMBER OR GROUPS FOR WHICH THE        *
C       * DATA IS REPRESENTATIVE.  THE DATA IS READ IN FROM THE FILE          *
C       * WHICH CONTAINS THE PROGRAM FORMATTED AS DESCRIBED ABOVE.            *
C       * THE INITIALIZATION MUST BE CHANGED FOR THE NUMBER OF GROUPS         *
C       * AND THE NUMBER OF OBSERVATIONS PER GROUP TO REFLECT THE DATA.       *
C       *                                                                     *
C       * WRITTEN BY LCDR CRAIG D. CARLSON IN PARTIAL FULFILLMENT OF THE      *
C       * REQUIREMENTS FOR A MASTER OF SCIENCE DEGREE IN SYSTEMS              *
C       * TECHNOLOGY (SPACE SYSTEMS OPERATIONS)*
C       *                                                                     *
C       * 27 SEPTEMBER 1985                                                   *
C       *                                                                     *
C       ***********************************************************************
C
C       *** VARIABLE DECLARATIONS ***
C
        DOUBLE PRECISION SUMX(15),SUMXSQ(15),XA,XE,TOTAL,R,N,X,Y,Z,FSTAT,
       *DF1,DF2
C
C       *** INITIALIZE THE NUMBER OF OBSERVATIONS AND THE NUMBER OF GROUPS
C
        R=15.D0
        N=33.D0
C
C       *** DO THE F-TEST FOR EACH OF THE SETS OF DATA PRESENT ***
C
        DO 100 J=1,3
C
C
        DO 10 I=1,15
           READ(5,20)SUMX(I),SUMXSQ(I)
20         FORMAT(1X,E23.16,2X,E23.16)
10      CONTINUE
```

```
C                                                                          FTE00450
      X=0.D0                                                               FTE00460
      Y=0.D0                                                               FTE00470
      Z=0.D0                                                               FTE00480
C                                                                          FTE00490
      DO 30 I=1,15                                                         FTE00500
         X=X+SUMX(I)                                                       FTE00510
         Y=Y+SUMXSQ(I)                                                     FTE00520
         Z=Z+SUMX(I)**2                                                    FTE00530
30    CONTINUE                                                             FTE00540
C                                                                          FTE00550
C                                                                          FTE00560
      XA=(Z/N)-(X**2/(R*N))                                                FTE00570
      XE=Y-(Z/N)                                                           FTE00580
      TOTAL=Y-(X**2/(R*N))                                                 FTE00590
C                                                                          FTE00600
      DF1=R-1.D0                                                           FTE00610
      DF2=R*(N-1.D0)                                                       FTE00620
C                                                                          FTE00630
      FSTAT=(XA/DF1)/(XE/DF2)                                              FTE00640
C                                                                          FTE00650
C     *** OUTPUT THE RESULTS-THE FORMATS REPRESENT VARIABLE DEFINITIONS FTE00660
C                                                                          FTE00670
      WRITE(6,31) J                                                        FTE00680
31    FORMAT('1',' F-STATISTICS FOR THE',I4,' COMPONENT')                  FTE00690
C                                                                          FTE00700
      WRITE(6,40) X                                                        FTE00710
40    FORMAT('0',' SUM OF SUMS =',E23.16)                                  FTE00720
C                                                                          FTE00730
      WRITE(6,41) Y                                                        FTE00740
41    FORMAT('0',' SUM OF SUMS**2 =',E23.16)                               FTE00750
C                                                                          FTE00760
      WRITE(6,42) Z                                                        FTE00770
42    FORMAT('0',' SUM OF SQUARE OF SUMS =',E23.16)                        FTE00780
C                                                                          FTE00790
      WRITE(6,50) XA                                                       FTE00800
50    FORMAT('0',' AMONG GROUP SUM OF SQUARES =',E23.16)                   FTE00810
C                                                                          FTE00820
      WRITE(6,51) XE                                                       FTE00830
51    FORMAT('0',' WITHIN GROUP SUM OF SQUARES =',E23.16)                  FTE00840
C                                                                          FTE00850
      WRITE(6,60) TOTAL                                                    FTE00860
60    FORMAT('0',' TOTAL SUM OF SQUARES =',E23.16)                         FTE00870
C                                                                          FTE00880
      WRITE(6,70) DF1                                                      FTE00890
70    FORMAT('0',' DEGREES OF FREEDOM IN NUMERATOR =',E23.16)              FTE00900
C                                                                          FTE00910
      WRITE(6,71) DF2                                                      FTE00920
71    FORMAT('0',' DEGREES OF FREEDOM IN DENOMINATOR =',E23.16)            FTE00930
C                                                                          FTE00940
```

```
      WRITE(6,80) FSTAT
80    FORMAT('0',' F-STATISTIC =',E23.16)
C
100   CONTINUE
C
      WRITE(6,81)
81    FORMAT('1')
      STOP
      END
```

## APPENDIX E

### RESULTS OF F-TEST

F-STATISTICS FOR THE 1 COMPONENT
SUM OF SUMS = 0.2091487882294994D 02
SUM OF SUMS**2 = 0.1810655469453404D 02
SUM OF SQUARE OF SUMS = 0.4128097510679048D 02
AMONG GROUP SUM OF SQUARES = 0.1211883136154515D 00
WITHIN GROUP SUM OF SQUARES = 0.1769374494346614D 02
TOTAL SUM OF SQUARES = 0.1781493325708159D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.7265063214387600D 00

F-STATISTICS FOR THE 2 COMPONENT
SUM OF SUMS = 0.2305792717805020D 02
SUM OF SUMS**2 = 0.1894701234015291D 02
SUM OF SQUARE OF SUMS = 0.4663825126655468D 02
AMONG GROUP SUM OF SQUARES = 0.1119371755000360D 00
WITHIN GROUP SUM OF SQUARES = 0.1848062982748736D 02
TOTAL SUM OF SQUARES = 0.1859256700298739D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.6424746465014737D 00

F-STATISTICS FOR THE 3 COMPONENT
SUM OF SUMS = 0.2693734535827635D 02
SUM OF SUMS**2 = 0.2124674921974186D 02
SUM OF SQUARE OF SUMS = 0.5921566381358867D 02
AMONG GROUP SUM OF SQUARES = 0.1084095881685183D 00
WITHIN GROUP SUM OF SQUARES = 0.2065459258160598D 02
TOTAL SUM OF SQUARES = 0.2076300216977449D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.5567362242775778D 00

```
F-STATISTICS FOR THE   4 COMPONENT
SUM OF SUMS = 0.317568863162706 5D 02
SUM OF SUMS**2 = 0.262110454230021 4D 02
SUM OF SQUARE OF SUMS = 0.807051309372626 6D 02
AMONG GROUP SUM OF SQUARES = 0.134718090369601 5D 00
WITHIN GROUP SUM OF SQUARES = 0.254039941136295 1D 02
TOTAL SUM OF SQUARES = 0.255387122039991 1D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.140000000000000 0D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.148500000000000 0D 04

F-STATISTIC = 0.562499748504186 9D 00
F-STATISTICS FOR THE   5 COMPONENT
SUM OF SUMS = 0.366910434953111 6D 02
SUM OF SUMS**2 = 0.338063374031769 5D 02
SUM OF SQUARE OF SUMS = 0.106968247335615 8D 03
AMONG GROUP SUM OF SQUARES = 0.172194024839614 8D 00
WITHIN GROUP SUM OF SQUARES = 0.327366549298207 9D 02
TOTAL SUM OF SQUARES = 0.329088489546604 1D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.140000000000000 0D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.148500000000000 0D 04
F-STATISTIC = 0.557933186678886 7D 00

F-STATISTICS FOR THE   6 COMPONENT
SUM OF SUMS = 0.416009787894843 1D 02
SUM OF SUMS**2 = 0.431323664202009 4D 02
SUM OF SQUARE OF SUMS = 0.140821535800736 0D 03
AMONG GROUP SUM OF SQUARES = 0.254454400511944 0D 00
WITHIN GROUP SUM OF SQUARES = 0.417241510621935 8D 02
TOTAL SUM OF SQUARES = 0.419786054627055 2D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.140000000000000 0D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.148500000000000 0D 04
F-STATISTIC = 0.646875756162344 2D 00
```

F-STATISTICS FOR THE 7 COMPONENT
SUM CF SUMS = 0.48103870923481110 02
SUM OF SUMS**2 = 0.5206409381611744D 02
SUM OF SQUARE OF SUMS = 0.1964653659903027D 03
AMONG GROUP SUM OF SQUARES = 0.4219987280210724D 00
WITHIN GROUP SUM OF SQUARES = 0.5009944015621441D 02
TOTAL SUM OF SQUARES = 0.5052143888423548D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.8934632362547178D 00

F-STATISTICS FOR THE 8 COMPONENT
SUM OF SUMS = 0.5623255941290002D 02
SUM CF SUMS**2 = 0.8590285622098203D 02
SUM OF SQUARE OF SUMS = 0.3070018365343552D 03
AMCNG GROUP SUM OF SQUARES = 0.9619512065933320D 00
WITHIN GROUP SUM OF SQUARES = 0.8283283785563848D 02
TOTAL SUM OF SQUARES = 0.8379478906223181D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.1231824736914027D 01

F-STATISTICS FOR THE 9 COMPONENT
SUM OF SUMS = 0.9931605295569187D 02
SUM OF SUMS**2 = 0.4043531056175881D 03
SUM OF SQUARE OF SUMS = 0.1416642417283150D 04
AMCNG GROUP SUM OF SQUARES = 0.7590638589699638D 01
WITHIN GROUP SUM OF SQUARES = 0.3901866814447566D 03
TOTAL SUM OF SQUARES = 0.3977773200344562D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.2063499133280512D 01

F-STATISTICS FOR THE 10 COMPONENT
SUM OF SUMS = 0.202994262878192 4D 03
SUM OF SUMS**2 = 0.1757564757819924D 04
SUM OF SQUARE OF SUMS = 0.7655692250451960D 04
AMONG GROUP SUM OF SQUARES = 0.4908580866354581D 02
WITHIN GROUP SUM OF SQUARES = 0.1681007835315404D 04
TOTAL SUM OF SQUARES = 0.1730093643978950D 04
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.3097309684192641D 01

F-STATISTICS FOR THE 11 COMPONENT
SUM OF SUMS = 0.2004773302247246D 03
SUM OF SUMS**2 = 0.1811875077561617D 04
SUM OF SQUARE OF SUMS = 0.7980439863108266D 04
AMONG GROUP SUM OF SQUARES = 0.5301029200839379D 02
WITHIN GROUP SUM OF SQUARES = 0.1732070678930534D 04
TOTAL SUM OF SQUARES = 0.1785080970938928D 04
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.3246332537532912D 01

F-STATISTICS FOR THE 12 COMPONENT
SUM OF SUMS = 0.2210834652773109D 03
SUM OF SUMS**2 = 0.2165859765218944D 04
SUM OF SQUARE OF SUMS = 0.9414279932820438D 04
AMONG GROUP SUM OF SQUARES = 0.6155753358218842D 02
WITHIN GROUP SUM OF SQUARES = 0.2071716965890740D 04
TOTAL SUM OF SQUARES = 0.2133274499472928D 04
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.3151731454585565D 01

F-STATISTICS FOR THE 13 COMPONENT
SUM OF SUMS = 0.1979583434471791D 03
SUM OF SUMS**2 = 0.1904701075347160D 04
SUM OF SQUARE OF SUMS = 0.7185446672840862D 04
AMONG GROUP SUM OF SQUARES = 0.4572946290150773D 02
WITHIN GROUP SUM OF SQUARES = 0.1832846608618751D 04
TOTAL SUM OF SQUARES = 0.1878576071520259D 04
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.2646478671459864D 01

F-STATISTICS FOR THE 14 COMPONENT
SUM OF SUMS = 0.9227736272614619D 02
SUM OF SUMS**2 = 0.2600827196886254D 03
SUM OF SQUARE OF SUMS = 0.9540726819081072D 03
AMONG GROUP SUM OF SQUARES = 0.3863985704619236D 01
WITHIN GROUP SUM OF SQUARES = 0.2505419928695444D 03
TOTAL SUM OF SQUARES = 0.2544059785741636D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.1635887377498235D 01

F-STATISTICS FOR THE 15 COMPONENT
SUM OF SUMS = 0.6880926336210145D 02
SUM OF SUMS**2 = 0.1018300362534556D 03
SUM OF SQUARE OF SUMS = 0.3695062863415098D 03
AMONG GROUP SUM OF SQUARES = 0.5385863804584070D 00
WITHIN GROUP SUM OF SQUARES = 0.9813497339004056D 02
TOTAL SUM OF SQUARES = 0.9867355977049896D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.5821433971075596D 00

F-STATISTICS FOR THE 16 COMPONENT
SUM OF SUMS = 0.6961387326450431D 02
SUM OF SUMS**2 = 0.1137157033969463D 03
SUM OF SQUARE OF SUMS = 0.3597053597878584D 03
AMONG GROUP SUM OF SQUARES = 0.3663260306209388D 00
WITHIN GROUP SUM OF SQUARES = 0.1101186497990677D 03
TOTAL SUM OF SQUARES = 0.1104849758296886D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.3528623485827816D 00

F-STATISTICS FOR THE 17 COMPONENT
SUM OF SUMS = 0.7052854165691613D 02
SUM OF SUMS**2 = 0.1548042004914454D 03
SUM OF SQUARE OF SUMS = 0.3820324148469269D 03
AMONG GROUP SUM OF SQUARES = 0.5041406896350342D 00
WITHIN GROUP SUM OF SQUARES = 0.1509838763429761D 03
TOTAL SUM OF SQUARES = 0.1514880170326111D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.3541763825767676D 00

F-STATISTICS FOR THE 18 COMPONENT
SUM OF SUMS = 0.7950452893326030D 02
SUM OF SUMS**2 = 0.1454127656122291D 03
SUM OF SQUARE OF SUMS = 0.5157161626920556D 03
AMONG GROUP SUM OF SQUARES = 0.9431815463208067D 00
WITHIN GROUP SUM OF SQUARES = 0.1402556039853086D 03
TOTAL SUM OF SQUARES = 0.1411987855316294D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.7133020797581566D 00

F-STATISTICS FOR THE 19 COMPONENT

SUM OF SUMS = 0.8206200282896035D 02

SUM OF SUMS**2 = 0.1642503769158388D 03

SUM OF SQUARE OF SUMS = 0.5788746875066006D 03

AMCNG GROUP SUM OF SQUARES = 0.1299298669532475D 01

WITHIN GROUP SUM OF SQUARES = 0.1584616300407728D 03

TOTAL SUM OF SQUARES = 0.1597609287103052D 03

DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02

DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04

F-STATISTIC = 0.8697276809711277D 00

F-STATISTICS FOR THE 20 COMPONENT

SUM OF SUMS = 0.9520048302882186D 02

SUM OF SUMS**2 = 0.2094086558414699D 03

SUM OF SQUARE OF SUMS = 0.8713219218815827D 03

AMCNG GROUP SUM OF SQUARES = 0.2671131239535162D 01

WITHIN GROUP SUM OF SQUARES = 0.2006954366226541D 03

TOTAL SUM OF SQUARES = 0.2033665678621893D 03

DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02

DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04

F-STATISTIC = 0.1411744637781583D 01

F-STATISTICS FOR THE 21 COMPONENT

SUM OF SUMS = 0.8994139952150146D 02

SUM OF SUMS**2 = 0.2335862791765991D 03

SUM OF SQUARE OF SUMS = 0.7245755635510632D 03

AMCNG GROUP SUM OF SQUARES = 0.1852785403586404D 01

WITHIN GROUP SUM OF SQUARES = 0.2263405235410884D 03

TOTAL SUM OF SQUARES = 0.2281933089446748D 03

DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02

DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04

F-STATISTIC = 0.8682828488687504D 00

F-STATISTICS FOR THE 22 COMPONENT
SUM OF SUMS = 0.1035648300735959D 03
SUM OF SUMS**2 = 0.2414810868436007D 03
SUM OF SQUARE OF SUMS = 0.9427341493765320D 03
AMONG GROUP SUM OF SQUARES = 0.2276892141650119D 01
WITHIN GROUP SUM OF SQUARES = 0.2320537453498354D 03
TOTAL SUM OF SQUARES = 0.2343306374914855D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.1040764077320930D 01

F-STATISTICS FOR THE 23 COMPONENT
SUM OF SUMS = 0.9031234717154746D 02
SUM OF SUMS**2 = 0.2047768116906234D 03
SUM OF SQUARE OF SUMS = 0.7149379755971323D 03
AMONG GROUP SUM OF SQUARES = 0.1711833054881912D 01
WITHIN GROUP SUM OF SQUARES = 0.1976274319346521D 03
TOTAL SUM OF SQUARES = 0.1993392649895340D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.9187822552243536D 00

F-STATISTICS FOR THE 24 COMPONENT
SUM OF SUMS = 0.9171706807335522D 02
SUM OF SUMS**2 = 0.1900438745495707D 03
SUM OF SQUARE OF SUMS = 0.8139562480340389D 03
AMONG GROUP SUM OF SQUARES = 0.2531548763025405D 01
WITHIN GROUP SUM OF SQUARES = 0.1819043120692303D 03
TOTAL SUM OF SQUARES = 0.1844358608322557D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.1476188171339999D 01

F-STATISTICS FOR THE 25 COMPONENT
SUM OF SUMS = 0.8440126623444366D 02
SUM OF SUMS**2 = 0.1782254895437147D 03
SUM OF SQUARE OF SUMS = 0.6892901073945840D 03
AMONG GROUP SUM OF SQUARES = 0.2143851912627546D 01
WITHIN GROUP SUM OF SQUARES = 0.1713325884697689D 03
TOTAL SUM OF SQUARES = 0.1734764403823964D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.1327251499840134D 01

F-STATISTICS FOR THE 26 COMPONENT
SUM OF SUMS = 0.6923579472492147D 02
SUM OF SUMS**2 = 0.1481985058366125D 03
SUM OF SQUARE OF SUMS = 0.4452980334185913D 03
AMONG GROUP SUM OF SQUARES = 0.1257250153391605D 01
WITHIN GROUP SUM OF SQUARES = 0.1437455255024266D 03
TOTAL SUM OF SQUARES = 0.1450027756558182D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.9277389287476910D 00

F-STATISTICS FOR THE 27 COMPONENT
SUM OF SUMS = 0.6366092822129744D 02
SUM OF SUMS**2 = 0.1177906819912730D 03
SUM OF SQUARE OF SUMS = 0.3935159041535181D 03
AMONG GROUP SUM OF SQUARES = 0.1233349853537058D 01
WITHIN GROUP SUM OF SQUARES = 0.1138555229497378D 03
TOTAL SUM OF SQUARES = 0.1150888728032749D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.1149027974258136D 01

```
F-STATISTICS FOR THE  28 COMPONENT
SUM OF SUMS = 0.5387592494868525D 02
SUM OF SUMS**2 = 0.1186133442790129D 03
SUM OF SQUARE OF SUMS = 0.2792595957915495D 03
AMONG GROUP SUM OF SQUARES = 0.8575190985312517D 00
WITHIN GROUP SUM OF SQUARES = 0.1158207483210974D 03
TOTAL SUM OF SQUARES = 0.1166782674196286D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.7853366268738306D 00

F-STATISTICS FOR THE  29 COMPONENT
SUM OF SUMS = 0.4536144151906700D 02
SUM OF SUMS**2 = 0.6564305139058856D 02
SUM OF SQUARE OF SUMS = 0.1919053830677748D 03
AMONG GROUP SUM OF SQUARES = 0.5472802462192574D 00
WITHIN GROUP SUM OF SQUARES = 0.6372399755991081D 02
TOTAL SUM OF SQUARES = 0.6427127780613007D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.9109723144851782D 00

F-STATISTICS FOR THE  30 COMPONENT
SUM OF SUMS = 0.3732568029970994D 02
SUM OF SUMS**2 = 0.5705765228634419D 02
SUM OF SQUARE OF SUMS = 0.1284820507967475D 03
AMONG GROUP SUM OF SQUARES = 0.3560162347433721D 00
WITHIN GROUP SUM OF SQUARES = 0.5577283177837672D 02
TOTAL SUM OF SQUARES = 0.5612884801312009D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.6770886363437515D 00
```

F-STATISTICS FOR THE 31 COMPONENT
SUM OF SUMS = 0.2973876286269883D 02
SUM OF SUMS**2 = 0.4217833610659068D 02
SUM OF SQUARE OF SUMS = 0.7726205204918280D 02
AMONG GROUP SUM OF SQUARES = 0.1830245094226048D 00
WITHIN GROUP SUM OF SQUARES = 0.4140571558609885D 02
TOTAL SUM OF SQUARES = 0.4158874009552145D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.4688645251806333D 00

F-STATISTICS FOR THE 32 COMPONENT
SUM OF SUMS = 0.2142367689889270D 02
SUM OF SUMS**2 = 0.1414820425416038D 02
SUM OF SQUARE OF SUMS = 0.3565723402182890D 02
AMONG GROUP SUM OF SQUARES = 0.5058971897285643D-01
WITHIN GROUP SUM OF SQUARES = 0.1379163191394209D 02
TOTAL SUM OF SQUARES = 0.1384222163291495D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.3890854828465455D 00

F-STATISTICS FOR THE 33 COMPONENT
SUM OF SUMS = 0.1732004830070321D 02
SUM OF SUMS**2 = 0.1115582097066481D 02
SUM OF SQUARE OF SUMS = 0.2312899538600082D 02
AMONG GROUP SUM OF SQUARES = 0.3130057176754676D-01
WITHIN GROUP SUM OF SQUARES = 0.1092453101680480D 02
TOTAL SUM OF SQUARES = 0.1095583158857235D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.3039120267386334D 00

F-STATISTICS OF MEAN VARIANCE, SKEWNESS AND KURTOSIS

F-STATISTICS FOR THE 1 COMPONENT
SUM OF SUMS = 0.111161743942250D 05
SUM OF SUMS**2 = 0.431942577749392D 07
SUM OF SQUARE OF SUMS = 0.1861260733960752D 08
AMONG GROUP SUM OF SQUARES = 0.3143833877400904D 06
WITHIN GROUP SUM OF SQUARES = 0.3755407373518861D 07
TOTAL SUM OF SQUARES = 0.4069790761258952D 07
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.4800000000000000D 03
F-STATISTIC = 0.2870223636519026D 01

F-STATISTICS FOR THE 2 COMPONENT
SUM OF SUMS =-0.3062348822555926D 07
SUM OF SUMS**2 = 0.7751235739069674D 12
SUM OF SQUARE OF SUMS = 0.2376364181020636D 13
AMONG GROUP SUM OF SQUARES = 0.5306562101878761D 11
WITHIN GROUP SUM OF SQUARES = 0.7031125381184634D 12
TOTAL SUM OF SQUARES = 0.7561781591372509D 12
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.4800000000000000D 03
F-STATISTIC = 0.2587626620217669D 01

F-STATISTICS FOR THE 3 COMPONENT
SUM OF SUMS = 0.1087692759875071D 08
SUM OF SUMS**2 = 0.1368235571001090D 14
SUM OF SQUARE OF SUMS = 0.3521258942662316D 14
AMONG GROUP SUM OF SQUARES = 0.8280430048704694D 12
WITHIN GROUP SUM OF SQUARES = 0.1261530754556778D 14
TOTAL SUM OF SQUARES = 0.1344335055043825D 14
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.4800000000000000D 03
F-STATISTIC = 0.2250444214596077D 01

## LIST OF REFERENCES


1. CLARKE, A.C., "Extra-Terrestrial Relays", Wireless World, vol. 51, no. 10, p. 305, Oct. 1945.

2. Pierce, J.R., "Orbital Radio Relays", Jet Propulsion, vol. 25, p. 153, Apr. 1955.

3. "Communication Satellites", Research Report, Encyclopedia Britannica, Inc., Chicago, Ill., 1978.

4. Jaffe, L., "NASA Communications Satellite Developments", Astronautics and Aerospace Engineering, vol. 1, no. 8, p. 48, Sep. 1963.

5. Edelson, B.I., Strauss, R. and Bargellini, P.L., "INTELSAT System Reliability", Acta Astronautica, vol. 2, p. 691, 1975.

6. Gagliardi, R.M., Satellite Communications, Lifetime Learning Publications, Belmont, Ca., 1984.

7. Bhargava, V.K., et al, Digital Communications by Satellite, John Wiley and Sons, Inc., 1981.

8. Feher, K., Digital Communications-Satellite/Earth Station Engineering, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1984.

9. Nyquist, "Certain Topics in Telegraph Transmission Theory", Transactions of the AIEE, Apr., 1928.

10. Shannon, et al, "The Philosophy of PCM", Proceedings of the IERE, Nov. 1948.

11. "Digital Signal Modulation Techniques in Satellite Communications", Research Report, Encyclopedia Britannica, Inc., Chicago, Ill., 1984.

12. Lender, A., "The Duobinary Technique for High Speed Data Transmission", IEEE Transactions on Communications and Electronics, vol. 82, p. 214, May, 1963.

13. Kabal, P. and Pasupathy, S., "Partial Response Signalling", IEEE Transactions on Communications, vol. COM-23, no. 9, Sep. 1975.

14. Oshita, S. and Feher, K., "Combined Effect of the Carrier Recovery and Symbol Timing Error on the Pe Performance of QPR and Offset QPR Systems", _IEEE Transactions on Communications_, vol. COM-30, no. 12, Dec. 1982.

15. Harmuth, H.F., _Sequency Theory, Foundations and Applications_, vol. 9, _Advances in Electronics and Electron Physics_, Academic Press, 1977.

16. Hahn, G.J. and Shapiro, S.S., _Statistical Models in Engineering_, John Wiley and Sons, Inc., 1967.

17. Hickman, E.P. and Hilton, J.G., _Probability and Statistical Analysis_, Intex Educational Publishers, 1971.

# BIBLIOGRAPHY


Ahmed, N. and Rao, K.R., *Orthogonal Transforms for Digital Signal Processing*, Springer-Verlag, New York, 1975.

Boutin, N. and Morisette, S., "Discrete Finite Duration Partial Response Signaling Pulse with Minimum Out of Band Power", *IEEE Globecom*, vol. 2, 1983.

Brigham, E.O., *The Fast Fourier Transform*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.

Feher, K., *Digital Communications Microwave Applications*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1981.

Herman, R., et al, "A Modular Program for Simulation of Digital Systems and Its Application to Satellite System Optimization", *Proceedings on the Joint Conference on Digital Processing of Signals in Communications*, IERE, Apr. 1972.

# INITIAL DISTRIBUTION LIST

| | | No. Copies |
|---|---|---|
| 1. | Defense Technical Information Center<br>Cameron Station<br>Alexandria, Virginia 22304-6145 | 2 |
| 2. | Superintendent<br>Attn: Library, Code 0142<br>Naval Postgraduate School<br>Monterey, California 93943-5100 | 2 |
| 3. | Commander<br>Naval Space Command<br>Dahlgren, Virginia 22448 | 1 |
| 4. | Professor Allen E. Fuhs, Code 72<br>Space Systems Academic Group<br>Naval Postgraduate School<br>Monterey, California 93943-5100 | 1 |
| 5. | Professor James L. Wayman, Code 53Ww<br>Department of Mathematics<br>Naval Postgraduate School<br>Monterey, California 93943-5100 | 7 |
| 6. | Space Systems Academic Group, Code 75<br>Naval Postgraduate School<br>Monterey, California 93943-5100 | 1 |
| 7. | LTCOL John T. Malokos, Code 39<br>Naval Postgraduate School<br>Monterey, California 93943-5100 | 1 |
| 8. | LCDR Craig D. Carlson<br>Attack Squadron 42<br>Oceana, Virginia 23460 | 6 |

214106

Thesis
C23253 Carlson
c.1 Computer simulation
of digital signal mod-
ulation techniques in
satellite communica-
tions.